

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-283276
(43)Date of publication of application : 03.10.2002

(51)Int.Cl. B25J 19/06
B25J 9/10
G05B 19/18

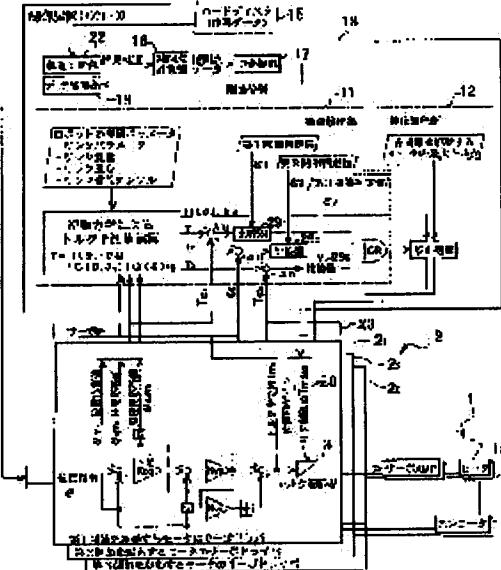
(21)Application number : 2001-081553 (71)Applicant : DAIHEN CORP
(22)Date of filing : 21.03.2001 (72)Inventor : KAMISHINA YASUHIRO

(54) COLLISION DETECTING-STOPPING CONTROL METHOD IN ARTICULATED ROBOT

(57)Abstract:

PROBLEM TO BE SOLVED: To highly accurately detect the occurrence of a collision in a short time from a fluctuation in joint driving torque generated by the collision, to relieve interference force from an obstacle at stopping processing time after detecting the collision, and to minimize damage.

SOLUTION: Torques T1, T2 and T3 to be generated by respective motors at collision nonoccurrence time are anticipated by an inverse dynamic operation from a position θ_m , a speed $\dot{\theta}_{dm}$, and acceleration $\ddot{\theta}_{dm}$ of the motors detected by servo drivers 21, 22 and 23 for controlling the respective joint driving motors 11, 12 and 13 of a robot. A difference ΔT between the torque anticipative value and a torque command value T_m actually generated by the servo drivers or the joint driving torque calculated from a torque response value is determined. When an absolute value of the difference ΔT is larger than a preset determining value ϵ , it is regarded that the collision is caused. After the collision, the joints are made to perform escape operation to reduce the interference force.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-283276

(P2002-283276A)

(43)公開日 平成14年10月3日(2002.10.3)

(51) Int.Cl. ⁷	識別記号	F I	データード(参考)
B 2 5 J 19/06		B 2 5 J 19/06	3 C 0 0 7
9/10		9/10	A 5 H 2 6 9
G 0 5 B 19/18		G 0 5 B 19/18	D X

審査請求 未請求 請求項の数12 O L (全 28 頁)

(21)出願番号	特願2001-81553(P2001-81553)	(71)出願人	000000262 株式会社ダイヘン 大阪府大阪市淀川区田川2丁目1番11号
(22)出願日	平成13年3月21日(2001.3.21)	(72)発明者	神品 泰宏 大阪市淀川区田川2丁目1番11号 株式会社ダイヘン内
		(74)代理人	100084593 弁理士 吉村 勝俊

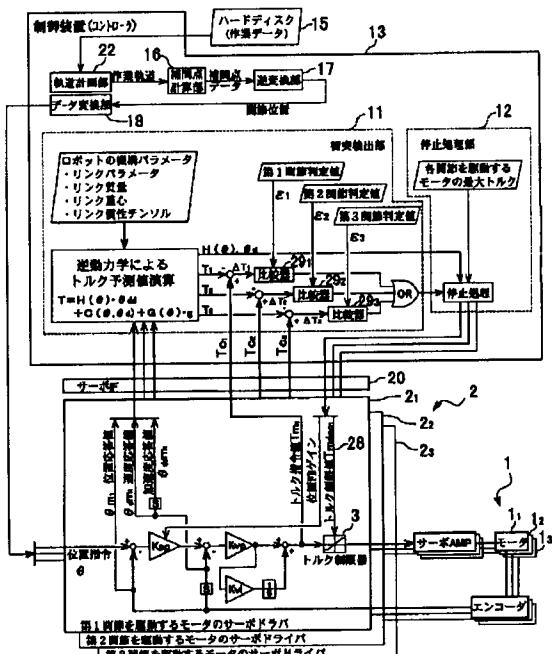
最終頁に続く

(54)【発明の名称】 多関節ロボットにおける衝突検出・停止制御法

(57)【要約】

【課題】 衝突により発生する関節駆動トルクの変動から衝突の発生を高精度かつ短時間に検出できること。さらには、衝突検出後の停止処理時に障害物からの干渉力を和らげまた損傷を最小限に抑えることができるようすること。

【解決手段】 ロボットの各関節駆動用モータ $1_1, 1_2, 1_3$ を制御するためのサーボドライバ $2_1, 2_2, 2_3$ が検出したモータの位置 θ_m ・速度 θ_d ・加速度 θ_{dd} から、衝突が発生していない場合に各モータが発生すべきトルク T_1, T_2, T_3 を逆運動力学演算によって予測する。そのトルク予測値と実際にサーボドライバが生成したトルク指令値 T_m あるいはトルク応答値から算出される関節駆動トルクとの差分 ΔT を求める。そして、差分 ΔT の絶対値が予め設定された判定値 ε より大きい場合に衝突が発生したものと見做すようにした。衝突後は、該当関節に逃げ動作させ、干渉力を軽減させるようにしている。



【特許請求の範囲】

【請求項1】複数の関節からなるロボットのリンクやリンク先端に把持されたツールまたはワークが周辺の障害物等に衝突したとき、リンク等の変形やロボット可動部の損傷を回避すべく衝突したことを検出しつつ直ちにロボットを停止させる方法において、

$$T = H(\theta) \cdot \dot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \cdot g$$

で示す式に基づく逆動力学演算により予測し、該トルク予測値と実際の関節駆動トルクとの差分を求め、

該差分の絶対値が予め設定された判定値より大きい場合に衝突が発生したものと見做すようにしたことを特徴とする多関節ロボットにおける衝突検出・停止制御法。

【請求項2】衝突が発生していない場合に各関節を駆動するモータが発生すべきトルクを予測するときに使用されるモータの位置・速度・加速度は、サーボドライバが検出したモータの位置・速度・加速度に代えて、サーボドライバに位置指令すべく予め教示されている作業データの再生動作により演算した関節位置と該関節位置をもとにして計算された速度および加速度であることを特徴とする請求項1に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項3】前記逆動力学演算は、ロボットの関節数をnとして、ロボットの各関節の加速度により各関節で発生するトルクならびに他の関節の加速度により発生する干渉トルクを示すn×nの慣性行列、遠心力とコリオリ力の影響を示すn×1の粘性行列、および重力加速度の影響により発生するトルクを示すn×3の重力行列を含むことを特徴とする請求項1または請求項2に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項4】前記n×nの慣性行列における対角要素としての「関節の持つ加速度が該関節自体に及ぼすトルクの関係を示した慣性モーメント」には、減速機を含むモータロータの慣性モーメントが減速比の2乗倍して加算され、各関節内における減速機を含むモータロータの慣性モーメントにより発生するトルクを考慮することができるようとしたことを特徴とする請求項3に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項5】前記逆動力学演算に供される式に、

【数2】

$$\left[\begin{array}{l} C_{m1} \cdot \theta_{dm1} + |F_{cou1}| \cdot \text{sgn}(\theta_{dm1}) \\ C_{m2} \cdot \theta_{dm2} + |F_{cou2}| \cdot \text{sgn}(\theta_{dm2}) \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ C_{mn} \cdot \theta_{dmn} + |F_{cou_n}| \cdot \text{sgn}(\theta_{dmn}) \end{array} \right]$$

が加算され、各関節内のモータ軸受や減速機の摩擦抵抗を考慮することができるようにしたことを特徴とする請求項1ないし請求項4のいずれか一項に記載された多関節ロボットにおける衝突検出・停止制御法。

ロボットの各関節駆動用モータを制御するためのサーボドライバが検出したモータの位置・速度・加速度から、衝突が発生していない場合に各モータが発生すべきトルクを、ロボットの各関節の位置・速度・加速度とロボット機構パラメータとをもとにした

【数1】

【請求項6】前記ロボットの機構パラメータは、各関節に連結されるリンクの重心位置・質量・長さであることを特徴とする請求項1ないし請求項5のいずれか一項に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項7】前記ロボットの機構パラメータには、慣性テンソルも付加されていることを特徴とする請求項6に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項8】前記実際の関節駆動トルクは、実際にサーボドライバが生成したトルク指令値から算出されるトルクであることを特徴とする請求項1ないし請求項7のいずれか一項に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項9】前記実際の関節駆動トルクは、各モータにおけるトルク応答値から算出されるトルクであることを特徴とする請求項1ないし請求項8のいずれか一項に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項10】複数の関節のうち停止もしくは低速で動作している関節あるいは慣性力が小さいため動作停止に必要なトルクが最大トルクと比較して十分小さい関節は、衝突があった後に衝突による干渉力を和らげるべく逃げ動作できるようにしたことを特徴とする請求項1ないし請求項9のいずれか一項に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項11】衝突発生時の各関節の速度と負荷イナーシャからロボットの動作を停止させるために必要な最小限のトルク値を関節ごとに算出し、その最小限のトルク値をサーボドライバのトルク制限器に設定すると共に全てのサーボドライバの位置フィードバック比例ゲインを0とし、強制的に速度指令値を0とした速度フィードバック制御による減速処理を行わせるようにしたことを特徴とする請求項10に記載された多関節ロボットにおける衝突検出・停止制御法。

【請求項12】ロボットの動作を停止させるために必要な各関節のトルク値の算出においては、全ての関節が同じ時間で停止すると仮定し、関節ごとに減速停止時間を算出した後、全ての減速停止時間のうちの最大値をいずれの関節の減速停止時間にも充てるようにしたことを特徴とする請求項10または請求項11に記載された多関節ロボットにおける衝突検出・停止制御法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は多関節ロボットにおける衝突検出・停止制御法に係り、詳しくは、ロボットが稼働中にワーク等の外界の対象物や障害物等に衝突すれば、それをいち早くかつ精度よく検出し、その後速やかにロボットの動きを止めたり力を解放して、ロボット本体やツールの変形、損傷を回避できるようにしたロボットの制御法に関するものである。

【0002】

【従来の技術】複数の関節からなるロボットのリンクやロボットの先端に把持されたツールもしくはワークが、教示データに基づいて再生動作しているときやティーチペンダントによる教示操作で動いているとき、意図することなく加工対象物や周辺の障害物に当たることがある。再生動作中であれば、ロボットの各関節を駆動するそれぞれのモータは指令値の通りに、また教示中でも衝突に気づかず教示を続ければ、ロボットは当然に障害物等を突破しても動こうとする。その結果、関節によっては過大なトルクが発生し、それに伴ってリンクやツールが変形したり関節等のロボットの可動部を損傷させるといった事態を招く。

【0003】ロボットに教示された作業データの再生動作を実行させる各モータを制御するためのそれぞれのサーボドライバには、例えば図14に示すように、位置や速度の制御系が組み込まれ、図示しない制御装置（ロボットコントローラ）から与えられる位置指令にモータ1の動作を追従制御させ、作業データの再生動作が実行されるようになっている。また、各サーボドライバ2には速度制御系の後にトルク制限器3が配置され、速度制御によって生成されるトルク指令を任意の値で制限しておくことができるようになっている。

【0004】この制御系から分かるように、一番外側のループである位置フィードバック系のエラー成分、すなわち位置指令から位置フィードバック値を差し引いた大きさ（位置の偏差）を監視すれば、衝突の有無を検出することができる。その最も基本的なものは、図15に示すように、位置制御ループにおける加算器4の直後からモータ1の位置をフィードバック制御した結果得られる位置偏差を取り出し、比較器5で「位置偏差のしきい値」である各関節における判定値6と比較して、大きければ衝突が発生したと判定するというものである。

【0005】このような制御によれば、ショックセンサトーチがぶつかるとスイッチが入りサーボ電源が落ちる仕組みになっていたり、機械式の衝突検出機構が装備されている検出装置など従来から存在するものに比べれば、ロボットとして必要最小限の構成要素であるモータさえ備えることで、すなわち衝突センサを有しなくてもロボットコントローラに衝突検出ソフトを搭載するだけで衝突を検出することが可能となる。従って、ロボット本体の機構の複雑化をきたさないようにしておくことが

できる。

【0006】衝突を検出する別の方法として、特許第2,749,724号に記載された外乱推定オブザーバを用いた衝突検出方法がある。これは、各サーボドライバの中に外乱トルクを観測するオブザーバを備え、実際にモータが回そうとしているイナーシャはどれくらいの大きさのものであるかを内部に持つモデルによって推定すると共に衝突により生じた外乱トルクを検出し、その外乱トルクの推定値に対する大小を見て衝突の有無を判定しようとするものである。

【0007】すなわち、内部でモデル化することにより或る指令値に対してどのような挙動をするかを予測し、その予測挙動と実際の挙動とを突き合わせることによって先読み補償するという思想である。従って、イナーシャがモデル化できていればトルク予測は可能であり、イナーシャに変動がなければ高い精度で衝突を検出することができるはずである。複数の関節を備えたモデル化はともかくとして、単一の関節だけで扱うとすればイナーシャのモデル化は容易であるからである。

【0008】このように、衝突の検出はぶつかったことによるサーボの状態変数の変化があって始めて可能となり、各モータ単位で検出されるトルクの変動によって衝突を感じているのである。

【0009】ところで、衝突を検出した後にロボットの動きを止める必要があるが、その停止のためにはいずれの関節においても可能なかぎりの大きさのトルクを発生させれば迅速に制動できるとの考え方から、トルク制限器には最大トルク値が設定されるのが一般的である。最大トルクとはそれぞれのモータにおいて発生することができる最大のトルクを意味していることは言うまでもない。

【0010】具体的には、位置フィードバックゲインを0とすることで速度制御系への指令を強制的に0とし、速度0を目標値とした速度フィードバック制御を行うことにより停止処理を行う。停止のための減速に使用するトルクは、速度制御系の後に配置されたトルク制限器により常にモータの最大許容トルクまで可能となっている。

【0011】

【発明が解決しようとする課題】ところが、前記した図15の衝突検出システムにおいては、いずれかの関節の位置のエラー成分が十分に蓄積されなければ、衝突を検出したと認識することができない。それがために、衝突を検出した時点では既にモータが大きなトルクを発生した状態となっており、ロボット本体やツールの変形・損傷を可及的に少なくすることはできない。すなわち、衝突を検出することはできても、その時点での遅れの状態になっているのである。

【0012】例示すれば、図2に示した第3関節 J_3 が第3リンク L_3 を破線のごとく真っ直ぐな姿勢に保持し

ようとしていても、例えば第2関節 J_2 によって押されれば障害物8に当たる。第3リンク L_3 は障害物8によって実線のように動かされようとするがフィードバック制御が効いているので、第3関節 J_3 を駆動しているモータ M_3 が直ちに最大トルクで真っ直ぐな位置へ戻そうとする。当然に干渉力は増大し、大きな力が相互に作用することになってしまう。

【0013】図15にも表されているように、サーボドライバー2にはトルク制御、その外側に速度制御、最外側に位置制御があるので、衝突に対して一番鋭敏なのはトルクとなる。その次に速度であり、一番鈍感なのが位置である。それゆえ、位置偏差で衝突を検出しようとすると、以下に述べる欠点は覆いようがない。

【0014】フィードバック制御においては速度やトルクを発生させるために位置偏差が常に存在しており、それが高速でしかも大きな加速度で動作するとき大きくなることは言うまでもない。一方、ゆっくりと動作する場合には、障害物に当たったときでも位置偏差が小さい。

【0015】そこで、小さな位置偏差で衝突であると判定させるようにしておくと、速い速度、大きい加速度で動作するときには小さな位置偏差を直ちに越えてしまうので衝突を誤検出することが多く、ロボットは止まってばかりで使い物にならなくなる。逆に、大きな位置偏差で衝突であると判定させるようにしておくと、遅い速度、小さな加速度で動くときには十分に行き過ぎなければ衝突を検出できなくなる。

【0016】これから分かるように、通常考えられる一番厳しい動かし方をしても衝突と誤認識しないだけの大きな設定値を入れざるを得なくなり、位置偏差をもとにした衝突の検出は鈍感なものとなってしまう。

【0017】特許第2,749,724号のように外乱推定オブザーバを使用した場合、そのオブザーバは一般的には速度ループに組み込まれる。これはサーボドライバにおいての話であるから各関節レベルで外乱を検出することになる。单一の関節だけで扱うとすればイナーシャのモデル化は容易であるが、複数の関節を持つ場合のモデル化は極めて難しく、多関節ロボットにおいては、衝突検出精度を十分に高く確保することが困難であると言える。

【0018】複数の関節からなるロボットでは、その関節を駆動する任意のモータと同じ速度パターンで動作させても、モータが発生するトルクは、ロボットの姿勢によって異なるという特性を持つ。ロボットの関節位置すなわちロボットの姿勢や各関節の速度・加速度の状態により、モータが駆動すべき負荷イナーシャやモータが受ける外乱トルクが変動してしまうためである。

【0019】言うまでもないが、ロボットは腕を伸ばせばイナーシャが大きくなり、短くすれば小さくなる。外乱推定オブザーバによる場合は高次の制御となるゆえ、複数の関節でオブザーバを定義することが現実には

不可能に近い。オブザーバの精度に要求される項目として、遠心力やコリオリ力等によっても生じる関節間の干渉が挙げられる。一つの関節でならこれらの速度の2乗項はたいした影響を持たないが、複数の関節からなる場合その影響は大きくなる。重力加速度による外乱トルクもロボットの姿勢が変われば変化する。回転半径が変わると重力によるモーメントも変わってくる。

【0020】衝突の検出は、ぶつかったことによるサーボの状態変数の変化があつて始めて可能となる。外乱推定オブザーバはサーボドライバに組み込まれるので外乱の検出はあくまでドライバレベルであり、従って衝突を感知するのは各モータ単位で検出されるトルクの変動であつて、それは各関節レベルでの外乱検出にとどまるところになる。

【0021】実際問題としてモータの先端にどのような制御対象物が存在するか分かっていないければならないが、一つの関節レベルで外乱トルクを検出するといつてもイナーシャ変動のない場合には正常に検出できても、イナーシャ変動のある場合やイナーシャ変動の避けられない多関節ロボットにおいては、そのトルク変動があつたとき何処までが起こるべくして起こったものか、それとも衝突によって起こったものは区別をつけがたい。すなわち、複数の関節があるとき上記した負荷イナーシャの変動に加えて、次に掲げる三つの外乱トルクが存在するからであり、その判定は難しい。

【0022】複数の関節からなるロボットの各関節が受ける外乱トルクの種類は、以下の三つである。一つは、他の関節の加速度運動より受ける干渉トルク（ロボットの関節位置・加速度により変化する）であり、二つ目は、遠心力やコリオリ力による外乱トルク（ロボットの関節位置・速度により変化する）である。そして、三つ目は、重力加速度による外乱トルク（ロボットの関節位置により変化する）である。

【0023】前述したごとく、トルクは衝突の検出において最も鋭敏であるが、これを検出の対象とする外乱推定オブザーバといえども、イナーシャ変動までは把握し得ない。なぜなら関節ごとに機能するからであり、他の関節を駆動するモータの状態と連動した上記の負荷イナーシャや外乱トルクの変動を考慮することができず、衝突によって発生した外乱トルクのみを精度よく抽出することは困難となる。このような事情を踏まえると、多関節ロボットの場合に厳密さを求めるにはまだ幾つかの課題が残されていると言わざるを得ない。

【0024】ところで、前述したごとく、衝突後にロボットを止める際速度0を目標とした速度フィードバック制御を行うことにしておりが、積分項が機能しているためにトルクは依然として発生し、それが大きくなることは避けられない。なぜなら、モータに最大トルクまで發揮させてもよいとしているのが一般的であり、モータが大きい力を発揮すれば、ロボットと障害物との間に大き

な力が掛かるることは必然的に起こるからである。すなわち、干渉があったときの逃げ動作が最大トルクで妨げられることになり、衝突による干渉力を全くと言ってよいほど軽減できなくなる問題がある。

【0025】ちなみに、衝突発生時にトルク制限器を0に設定することはできない。ぶつかっているといえども慣性力があるので、慣性力に対して制動するのは摩擦だけということになり、制動力が著しく低下する。それだけでなく、停止している関節についても、衝突による衝撃力や他の関節から受ける干渉トルクによる関節の運動を制動できなくなるからである。

【0026】本発明は上記した問題に鑑みなされたもので、その目的は、大きく分けて以下に記す二つの課題を克服する多関節ロボットにおける衝突検出・停止制御法を提供することである。第一の課題は、関節駆動トルクが非線形的に変動する複数の関節からなるロボットにおいても、衝突により発生する関節駆動トルクの変動から衝突が発生したことを高精度かつ短時間に検出することである。すなわち、衝突を感じているのは各モータ単位で検出されるトルクの変動であるが、できるだけ厳密な計算をして衝突によるものか否かを見極めることができるようにしようとするのである。

【0027】第二の課題は、ロボットの衝突検出後の停止処理において、衝突による障害物からの干渉力を和らげ、ロボット本体やぶつかった対象物の損傷を最小限に抑えることである。言い換えれば、大きな慣性力を持っていない関節もしくは速度が発生していない関節については、力方向に逃げる仕組みを提案しようとするものである。

【0028】

【課題を解決するための手段】本発明は、複数の関節からなるロボットのリンクやリンク先端に把持されたツールまたはワークが周辺の障害物等に衝突したとき、リンク等の変形やロボット可動部の損傷を回避すべく衝突したことを検出しつつ直ちにロボットを停止させる方法に適用される。その特徴とするところは、図1を参照して、ロボットの各関節駆動用モータ $1_1, 1_2, 1_3$ を制御するためのサーボドライバ $2_1, 2_2, 2_3$ が検出したモータの位置 θ_m ・速度 $\theta_d m$ ・加速度 $\theta_{dd} m$ から、衝突が発生していない場合に各モータが発生すべきトルク T_1, T_2, T_3 を、ロボットの各関節の位置・速度・加速度とロボット機構パラメータとをもとにした

【数3】

$$T = H(\theta) \cdot \theta_{dd} + C(\theta, \theta_d) + G(\theta) \cdot g$$

で示す式に基づく逆動力学演算により予測する。そのトルク予測値と実際の関節駆動トルク T_c との差分 ΔT を求める。そして、差分 ΔT の絶対値が予め設定された判定値 ϵ より大きい場合に衝突が発生したものと見做すようにしたことである。

【0029】衝突が発生していない場合に各関節を駆動

するモータが発生すべきトルクを予測するとき使用されるモータの位置・速度・加速度として、図9に示すように、サーボドライバ $2_1, 2_2, 2_3$ に位置指令すべく予め教示されている作業データの再生動作により演算した関節位置 θ と、その関節位置をもとにして計算された速度 θ_d および加速度 θ_{dd} を採用してもよい。

【0030】逆動力学演算においては、ロボットの関節数をnとすれば、ロボットの各関節の加速度により各関節で発生するトルクならびに他の関節の加速度により発生する干渉トルクを示す $n \times n$ の慣性行列、遠心力とコリオリ力の影響を示す $n \times 1$ の粘性行列、および重力加速度の影響により発生するトルクを示す $n \times 3$ の重力行列が含まれる。

【0031】 $n \times n$ の慣性行列における対角要素としての「関節の持つ加速度が該関節自体に及ぼすトルクの関係を示した慣性モーメント」には、減速機を含むモータロータの慣性モーメントが減速比の2乗倍して加算され、各関節内における減速機を含むモータロータの慣性モーメントにより発生するトルクを考慮できるようにしている。

【0032】逆動力学演算に供される式に、

【数4】

$$\left| \begin{array}{l} C_{m1} \cdot \theta_{dm1} + |F_{cou1}| \cdot \text{sgn}(\theta_{dm1}) \\ C_{m2} \cdot \theta_{dm2} + |F_{cou2}| \cdot \text{sgn}(\theta_{dm2}) \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ C_{mn} \cdot \theta_{dmn} + |F_{cou_n}| \cdot \text{sgn}(\theta_{dm_n}) \end{array} \right|$$

を加算しておき、各関節内のモータ軸受や減速機の摩擦抵抗を考慮することができるようにしておくとよい。

【0033】ロボットの機構パラメータには、各関節に連結されるリンクの重心位置・質量・長さが含まれていればよい。さらに、慣性テンソルも付加されているならば、より一層好ましい。

【0034】トルク予測値との差分を求めるときの実際の関節駆動トルクには、実際にサーボドライバが生成したトルク指令値から算出されるトルクを充てておけばよい。各モータにおけるトルク応答値から算出されるトルクを採用すれば、衝突検出精度はより一層高められる。

【0035】関節のうち停止もしくは低速で動作している関節あるいは慣性力が小さいため動作停止に必要なトルクが最大トルクと比較して十分小さい関節には、衝突があった後に衝突による干渉力を和らげるべく逃げ動作できるようにしておく。

【0036】ロボットの停止処理をするにあたり、衝突発生時の各関節の速度と負荷イナーシャからロボットの動作を停止させるために必要な最小限のトルク値を関節ごとに算出し、その最小限のトルク値をサーボドライバのトルク制限器3に設定すると共に全てのサーボドライバの位置フィードバック比例ゲインKppを0とし、強制

的に速度指令値を0とした速度フィードバック制御による減速処理を行わせるようとする。

【0037】ロボットの動作を停止させるために必要な各関節のトルク値の算出にあたっては、全ての関節が同じ時間で停止すると仮定して、関節ごとに減速停止時間 t_{md_1} , t_{md_2} , t_{md_3} を算出したうえで、全ての減速停止時間のうちの最大値をいずれの関節の減速停止時間にも充てるようすることができる。

【0038】

【発明の実施の形態】以下に、本発明に係る多関節ロボットにおける衝突検出・停止制御法を、その実施の形態を表した図面を参考しながら詳細に説明する。まず、本発明が適用される例としてのアーク溶接ロボットの構成の概略、ならびにリンクを連結している関節を駆動するモータの制御から述べる。なお、本発明は、ロボットが作業中にワーク等の外界の対象物やその他の障害物に衝突したとき、いち早くかつ精度よく衝突のあったことを検出し、以後速やかにロボットの動きを止めたりリンクに作用する力の全部もしくは一部を解放して、ロボット本体やツールもしくはワークの変形や損傷等を軽減または回避できるようにしようとするものである。

【0039】詳細は図や表をもとにして後述するが、複数の関節からなるロボットにおけるいずれかのリンクの逃げ動作を概略的に示す図2を参照して、本発明の基盤となるところを説明する。この例は三つの関節 7_1 , 7_2 , 7_3 からなるロボット10であり、第1関節 7_1 および第2関節 7_2 の高速動作中に第3リンク L_3 が障害物8に衝突した場合を表している。述べるまでもないが、図中の矢印9のように、停止しているか動きの少ない第3関節 7_3 が衝突による干渉力を和らげる方向に回転すれば、ロボット全体としての衝突による衝撃をおおい緩和することができる。

【0040】ところで、従来技術の項でも触れたが、関節駆動トルクが非線形的に変動する複数の関節からなるロボットにおいては、衝突が発生していないときでも以下の要因を考慮して、各関節が発生するトルク値を正確に予測しておくことが重要である。それらをもとにしても、衝突が原因で発生する関節駆動トルクの変動を検出すれば、衝突の発生したことを高精度かつ短時間に検出することができるようになるからである。

【0041】その要因とは、(1) ロボット各関節の位置の変化により生じる負荷イナーシャの変動(「イナーシャ変動」という)、(2) ロボット各関節の位置・加速度の変化により生じる他の関節の加速度運動から受ける干渉トルクの変動(「加速度の干渉」という)、(3) ロボット各関節の位置・速度の変化により生じる遠心力やコリオリ力からなる外乱トルクの変動(「速度の干渉」という)、(4) ロボット各関節の位置変化により生じる重力加速度の影響による外乱トルクの変動(「重力の干渉」という)の四つである。

【0042】本発明の最初の例を説明する制御ブロック線図は、図1に表されている。複数の関節を持つ図2に示したロボット10のリンク L_1 , L_2 , L_3 やリンク先端に把持されたツールまたはワーク(図示せず)が、周囲の障害物8に衝突したときリンクやツールの変形を防止しさらには関節等のロボット可動部の損傷を回避すべく、図1中の衝突検出部11で衝突したことを検出し、かつ直ちに停止処理部12においてロボットを停止させることができるようになっている。なお、3つの関節それぞれのためにモータ 1_1 , 1_2 , 1_3 が装備され、そのモータごとにロボットコントローラ13からの指令を受けて動作するサーボドライバ 2_1 , 2_2 , 2_3 が準備されている。

【0043】本発明に係る制御における衝突検出処理の主たるところは、以下の工程からなる。まず、ロボットの関節駆動用の各モータを制御するサーボドライバ 2_1 , 2_2 , 2_3 が検出したモータの位置 θ_{m_1} , θ_{m_2} , θ_{m_3} 、速度 $\theta_d m_1$, $\theta_d m_2$, $\theta_d m_3$ 、加速度 $\theta_{dd} m_1$, $\theta_{dd} m_2$, $\theta_{dd} m_3$ から、衝突が発生していない場合に各モータが発生しているはずのトルク T_1 , T_2 , T_3 を、後述する逆動力学演算によって予測する。

【0044】次に、そのトルク予測値 T_1 , T_2 , T_3 と、実際の関節駆動トルク、すなわちサーボドライバ 2_1 , 2_2 , 2_3 が実際に生成したトルク指令値 T_{m_1} , T_{m_2} , T_{m_3} から算出されるトルクとの差分 ΔT_1 , ΔT_2 , ΔT_3 を求める。そして、いずれかの差分 ΔT_i の絶対値が予め設定された該当する判定値 ε_i より大きければ、衝突が発生したものと見做すようにしている。

【0045】衝突の検出は、ぶつかったことによるサーボの状態変数の変化があつて始めて可能となることは、既に述べた。また、衝突を感じているのは各モータ単位で検出されるトルクの変動であることも触れた。そのトルクの変動が起こるべくして起こったものが衝突によって起こったものは、複数の関節があるとき上で述べた四つの外乱トルクの存在を見極めなければ判断することができない。しかしながら、その存在を見極めるためには、厳密な計算が求められることになる。

【0046】そこで、ロボットが所望の運動をするために必要な各関節に発生させるべきトルクを求め、これを実際にモータが発生しているトルクと突き合わせ、その違いから衝突の有無を検出することにする。演算量が多くなる傾向はあるが、検出はより一層正確なものとなる。この例では、モータが励磁された後は何時でもサーボドライバからモータの現在位置・速度・加速度を取り込み、それらで各関節が動いているとして各モータに出ているはずのトルク値を計算するというものである。

【0047】ここで、まず図3を参照して、ロボットシステムの構成について説明する。ロボットは用途に応じ

て適宜の数の関節を備えるが、図の簡潔化を図るために既に示したとおり、3つの関節からなるロボットを例にする。ロボットの各関節を駆動するモータ $1_1, 1_2, 1_3$ のそれには、回転による前後進量としての位置情報を提供するエンコーダ $1e_1, 1e_2, 1e_3$ が取り付けられている。

【0048】これらのモータを駆動するための制御装置(ロボットコントローラ)13はロボット本体とは別の制御ボックスに配備され、ティーチペンダント14で指令された教示データを記憶し、その記憶データと予め設定記憶されている当該ロボット固有の機構パラメータ等とをもとに動作を再生させる演算が行われるようになっている。

【0049】そして、コントローラ13とモータ1との間には、各モータを位置・速度制御するサーボドライバ $2_1, 2_2, 2_3$ や、位置・速度制御により求められた指令電流値をもとにモータに電力を供給するサーボアンプ $2a_1, 2a_2, 2a_3$ も設けられる。ちなみに、サーボドライバ2の構成は従来の技術の項で述べた図14のとおりであり、これが図1の下半部に記載しているかたちで搭載されている。

【0050】各サーボドライバには、制御装置から与えられる位置指令にモータを追従制御させるため、位置制御ループと速度制御ループとが組み込まれている。位置制御ループは比例要素によるフィードバック制御が、速度制御ループについては比例要素および積分要素によるフィードバックが行われる。なお、位置フィードバック比例ゲインKppは、ロボットコントローラからの要求に応じて、任意の値に変更することができる。

【0051】速度制御系の後にはトルク制限器3が配置され、速度制御により生成されるトルク指令値Tm_iを任意の値で制限しておくことができるようになっている。トルク制限器3に設定される制限値も、コントローラ13からの要求に応じて任意の値に変更可能である。なお、図中のKvpは速度フィードバック比例ゲイン、Kviは速度フィードバック積分ゲイン、Sは微分要素、1/Sは積分要素を表している。

【0052】コントローラ13は3つの関節に対する制御指令を発するもので、大別すると図3に表したように、中央処理装置CPU、演算プログラムや各種データを記憶して演算する記憶演算部、教示データの取り込みや演算値の出力をするためのインターフェイスといったものからなる。ここで、教示されたデータ(以下作業データという)の再生動作を、以下に簡単に述べる。

【0053】制御プログラムはROMに格納されており、その中の軌道計画部22が図1に示すようにハードディスク15に格納されている作業データを読み出し、ロボット先端に把持されるツールまたはワークの作業軌道が三次元空間の直交座標上で計画される。

【0054】次いで、補間点計算部16がその作業軌道

を予め規定された時間の補間周期ごとに分割し、補間周期ごとにツール等が到達すべき位置・姿勢を補間点データとして計算する。そして、逆変換部17が補間点データ中の直交座標で表された位置・姿勢を、多関節ロボットで再現するに必要な各関節レベルでの位置に逆変換し、各サーボドライバ $2_1, 2_2, 2_3$ への位置指令値を演算する。

【0055】このサーボドライバへの位置指令はRAMに一時的に記憶され、ロボットの関節位置をサーボ位置指令値にデータ変換するデータ変換部18から、同期信号S_{YNC}に同期して各サーボドライバへ伝達される。なお、以上の処理は、多関節ロボットをティーチングプレイバック方式で制御するうえで最も基本的な処理であり、公知であることは言うまでもない。

【0056】ちなみに、図3中のROMには、ロボットの動作制御ならびに衝突検出や停止処理を実行するプログラムと、その実行条件や制御定数などのデータが格納される。RAMはCPUのワーキングエリアとして用いられ、計算途中のデータを一時的に保存する。ハードディスク15には作業データのみならず、ロボット固有のデータ例えばリンクの長さ・質量といったものや、教示された動作において固有のものであるワークやツールに関するデータ等が格納される。

【0057】シリアルI/O(インプット・アウトプット)19はティーチペンダント14と接続され、ティーチペンダントの操作による作業データがシリアル通信によってCPUで認識できるようになっている。サーボIF(インターフェイス)20はサーボドライバ $2_1, 2_2, 2_3$ に接続され、指令データの送信やモータの制御状態のモニタを可能にする。タイマ21は、定期時刻ごとに同期信号S_{YNC}を発生し、この信号によってサーボドライバへの指令値の更新タイミングが与えられる。

【0058】次に、本発明によるロボットの衝突検出ならびに衝突後の停止処理を行うための最初の形態例を、図1に基づいて説明する。なお、停止処理は後述することにして、衝突検出から述べる。衝突の検出における基本的なところは、比較器 $29_1, 29_2, 29_3$ における比較の対象を、位置偏差ではなくトルクとしている点である。

【0059】ところで、トルクをもとに衝突の有無を検出しようとする場合、大きいトルクが掛かっていても一概に衝突しているとは言えないことに留意しなければならない。関節が大きな加速度で動くときは大きなトルクが要求されるので、トルクが急激に大きくなったからといって衝突と見做すわけにはいかない。よって、本来所望する運動をさせるために必要なトルクは幾らか、今出ているトルクとはどの程度かけ離れているかを厳密に判定しなければ、高精度な衝突検出はできなくなる。本発明はこのような観点から研究を重ね、完成したものである。

【0060】まず、サーボドライバはモータの駆動を制御するだけでなく、サーボモータの状態をモニタできるものであるから、本例においてはサーボドライバからモータの位置・速度・加速度といったモータ情報を取り出すことにする。そして、これをコントローラ13の衝突検出部11に入力して、後で詳しく述べる $H(\theta) \cdot \theta_{dd} + C(\theta \cdot \theta_d) + G(\theta) \cdot g$ なる計算式をもとにした逆動力学演算を行い、衝突していないければ各関節駆動用のモータが発生するはずのトルク T_1, T_2, T_3 を予測する。

【0061】逆動力学演算とは、ロボットの各関節の位置・速度・加速度の状態と、ロボットの機構パラメータから、ロボットの各関節に発生するトルクを算出するものである。この逆動力学演算によれば、前記した(1)から(4)のトルク変動要因の影響を正確に予測することができるようになる。

【0062】一方、予測トルクと比較される対象の「実際に発生しているトルク」は、サーボドライバのトルクループの手前で拾い上げられたトルク値である。衝突がなければ、逆動力学で演算されたトルク値とほとんど同じであり、両トルクの差が大きければ衝突していることを意味することになる。従って、その差分が関節ごとに予め決められた後述する「しきい値」 e_i に達していれば、「衝突した」と見做すことになるのである。

【0063】 $T_i = H(\theta) \cdot \theta_{dd} + C(\theta \cdot \theta_d) + G(\theta) \cdot g$ を計算するにあたり、そのロボット固有のデータすなわち機構パラメータも必要であり、それらがハードディスクから読み出される。トルク値は電流値でも $N \cdot m$ (ニュートン・メータ) でもどちらでもよい。サーボドライバの中にトルク定数 ($N \cdot m/A$) を持つ

たトルク換算部があれば $N \cdot m$ で出力することができ、それを演算値と突き合わせればよい。サーボドライバにトルク換算部がなければ、逆動力学演算において電流値に換算して比較すればよいことは言うまでもない。

【0064】各サーボドライバとロボットコントローラとの間にはサーボIF20が設置され、以下のデータが伝達される。図1に示されているように、サーボドライバ2からコントローラ13へは、モータの位置応答 θ_m [rad]、モータの速度応答 θ_{dm} [rad/sec]、モータの加速度応答 θ_{ddm} [rad/sec²] およびモータへのトルク指令値 T_m [N·m] であり、コントローラからサーボドライバへは、位置指令値 θ [rad]、位置フィードバック比例ゲイン K_{pp} および後述するトルク制限値 T_{mdec} [N·m] である。

【0065】衝突の検出は、サーボドライバのフィードバック制御が行われている限り、図4に示す5つのステップが順次繰り返される。まず、ステップ1(図面では以下S1などと記す)では、衝突検出部11がサーボIF20を介してサーボドライバ2から、ロボットの各関節を駆動するモータの位置・速度・加速度の応答値とトルク指令値を読み込む。この読み込みは、関節を駆動する全てのモータに対して行われる。3つの関節からなるロボットの場合では、モータの位置・速度・加速度の応答値ならびにトルク指令値は、表1の記号を用いて、それぞれ式(1-1)ないし(1-4)で表される行列となる。なお、式中の記号の添え字は関節番号を表し、[]^T は表1の「個数」の欄に記載された行列を「転置」して表示したこと示している。

【表1】

記号	項目	単位	個数
θ_m	モータの位置応答	rad	$n (= 3) \times 1$ 行列
θ_{dm}	モータの速度応答	rad/sec	$n (= 3) \times 1$ 行列
θ_{ddm}	モータの加速度応答	rad/sec ²	$n (= 3) \times 1$ 行列
T_m	モータのトルク指令値	N·m	$n (= 3) \times 1$ 行列

【数5】

$$\theta_m = [\theta_{m1}, \theta_{m2}, \theta_{m3}]^T \quad (1-1)$$

$$\theta_{dm} = [\theta_{dm1}, \theta_{dm2}, \theta_{dm3}]^T \quad (1-2)$$

$$\theta_{ddm} = [\theta_{ddm1}, \theta_{ddm2}, \theta_{ddm3}]^T \quad (1-3)$$

$$T_m = [T_{m1}, T_{m2}, T_{m3}]^T \quad (1-4)$$

【0066】ちなみに、実施例では関節数を3としているが、本発明の趣旨からして $n \geq 2$ で適用することができる。 $n = 1$ の場合も理屈上は不可能でないが、低自由度のロボットは通常速度は速くなくまた他関節からの干渉も受けないので、行列演算の必要性はない。なお、例えば一方にはツールが装着され他方にワークが把持され

た2基のロボットが連動して再生動作するシステムの場合であっても、本発明の思想は各ロボットごとに適用されるものである。

【0067】ところで、モータとロボット関節とは、減速機を介して結合されるのが一般的である。そのため、式(1-1)ないし(1-4)の値を、ロボット関節での位置・

速度・加速度に換算する必要がある。関節における位置
・速度・加速度の応答値ならびに駆動トルクは、表2の
記号を用いてそれぞれ式(1-5)ないし(1-8)で表すこと

ができる。

【表2】

記号	項目	単位	個数
θ_r	ロボット関節の位置応答	rad	$n (= 3) \times 1$ 行列
θ_{dr}	ロボット関節の速度応答	rad/sec	$n (= 3) \times 1$ 行列
θ_{ddr}	ロボット関節の加速度応答	rad/sec ²	$n (= 3) \times 1$ 行列
T_c	ロボット関節の駆動トルク	N·m	$n (= 3) \times 1$ 行列
ζ	ロボット関節の減速比	—	$n (= 3) \times 1$ 行列

【数6】

$$\theta_r = [\theta_{r1}, \theta_{r2}, \theta_{r3}]^T \quad (1-5)$$

$$\theta_{dr} = [\theta_{dr1}, \theta_{dr2}, \theta_{dr3}]^T \quad (1-6)$$

$$\theta_{ddr} = [\theta_{ddr1}, \theta_{ddr2}, \theta_{ddr3}]^T \quad (1-7)$$

$$T_c = [T_{c1}, T_{c2}, T_{c3}]^T \quad (1-8)$$

$$\zeta = [\zeta_1, \zeta_2, \zeta_3]^T \quad (1-9)$$

【0068】第*i*関節を例にすれば、それらは式(1-10)ないし(1-13)で表される。位置・速度・加速度については減速比 ζ_i で除した値が関節換算値となり、トルクにおいては減速比を乗じた値が関節換算値となる。そこ

$$\theta_{ri} = \theta_{mi}/\zeta_i \quad (1-10)$$

$$\theta_{dr_i} = \theta_{dm}/\zeta_i \quad (1-11)$$

$$\theta_{ddr_i} = \theta_{ddm}/\zeta_i \quad (1-12)$$

$$T_{ci} = T_{mi} \cdot \zeta_i \quad (1-13)$$

【0069】関節レベルの位置・速度・加速度ならびにトルクの応答値が分かれれば、ステップ3においては、これらを次に述べる逆動力学モデル(H , C , G)に適用した逆動力学演算によって、発生すると思われるトルクが計算される。逆動力学モデルには、慣性行列の H 行列と粘性行列の C 行列と重力行列の G 行列とが含まれている。なお、各行列の物理的意味は後述する。

で、ステップ2においては、モータレベルの位置・速度・加速度がロボットの関節レベルの値に変更される。

【数7】

$$T = H(\theta) \cdot \theta_{dd} + C(\theta, \theta_d) + G(\theta) \cdot g \quad (2-1)$$

【0071】式(2-1)は、広瀬茂男著「ロボット工学－機械システムのベクトル解析－」裳華房、第2版、1989年2月10日発行（以下技術文献1という）に記載されている。その節11.3の〔4〕には、剛体の運動解析手法の一つであるニュートン・オイラー法を用いて逆動力学演算する手順が詳しく説明されているので、ここでは説明を省く。最終的に逆動力学演算式は技術文献1中の式(11.28)となるが、その式中の最後の項 $J^T K$ （転置したヤコビ行列にロボットの先端に掛かる外力と外力によるモーメントを表すKを乗じたもの）は、本発明の場合、ロボットが衝突していない状態、すなわち空間を何物にも触れずに変位しているときのトルクを求めようと

【0070】ロボットの機構パラメータ（関節に連結されるリンクの重心位置・質量・慣性テンソル・長さ等）と、各関節の位置・速度・加速度から、各関節に発生するトルク T は式(2-1)により演算できることが知られている。これを逆動力学モデルという。

【数8】

することから考慮する必要がなく、従ってその項を省くことによって上記の式(2-1)を得ることができる。

【0072】関節数nのロボット場合、慣性行列Hはn×nの正則な対称行列であり、ロボットの各関節の加速度によって関節自体に発生するトルクならびに他関節の加速度によって発生する干渉トルクを意味する。粘性行列Cはn×1行列であり、ロボットの各関節の速度により生じる遠心力およびコリオリ力の影響で生じるトルクを示し、重力行列Gはn×3行列で、重力加速度の影響により発生するトルクを示すものである。

【0073】慣性行列Hについて、少し詳しく触れる。慣性行列H（後述する式(2-9)を参照）の対角要素のH

J_{jj} は、第 j 関節の加速度が第 j 関節自体に及ぼすトルクの関係を示す慣性モーメントを表しており、すなわち H_{jj} は関節位置により変動する負荷イナーシャとなる。その他の成分である H_{ij} は、第 j 関節の加速度が第 i 関節に及ぼす干渉トルクの関係を表す慣性モーメントである。

【0074】慣性行列 H 、粘性行列 C および重力行列 G は、ロボットの機構パラメータと各関節の位置・速度か

ら計算できる。ロボットの機構パラメータは、以下の4つから構成される。なお、表3中の個数は、関節に連結されるリンク 1 つ当たりである。ちなみに、図2に表したロボットの場合の第3リンク L_3 に溶接トーチ等のツールが装着されているならば、その第3リンクのパラメータはツールが一体となった状態で与えられることになる。

【表3】

記号	項目	単位	個数
m	関節に連結されるリンクの質量	kg	1
m_c	関節に連結されるリンクの重心位置 $m_c = [m_x, m_y, m_z]$	m	1×3
I	関節に連結されるリンクの慣性テンソル $I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$	kg · m ²	3×3 行列
a	関節に連結されるリンクの長さ	m	1

【0075】各リンクの質量・重心位置・慣性テンソル・長さ等のロボット機構パラメータは、ロボットの機構やリンク形状および材質から求めることが可能な既知の値であり、それがロボットの運動中に変化しないことも明らかであるから固定値であって、予めハードディスク 15 の中に記憶される。それらの機構パラメー

タを読み出して式(2-1)に適用すれば、各関節の現在の位置・速度・加速度の3つと重力加速度 g とから、各関節に発生するトルクを演算することができる。なお、式(2-1)の各要素は、表4に記載の意味を持つ。

【表4】

記号	項目	単位	個数
T	関節に発生するトルク	N · m	$n (= 3) \times 1$ 行列
θ	ロボット関節の位置応答	rad	$n (= 3) \times 1$ 行列
$\dot{\theta}_i$	ロボット関節の速度応答	rad/sec	$n (= 3) \times 1$ 行列
$\ddot{\theta}_i$	ロボット関節の加速度応答	rad/sec ²	$n (= 3) \times 1$ 行列
H	慣性行列	kg · m ²	$n (= 3) \times n$ 行列
C	粘性行列	N · m	$n (= 3) \times 1$ 行列
G	重力行列	kg · m	$n (= 3) \times 3$ 行列
g	重力加速度ベクトル	m/sec ²	3×1 行列
n	ロボットの制御関節数	—	—

【0076】ここで、本例におけるロボットの機構パラメータについて説明を加える。質量 m_i 、重心位置 $m_{x_i}, m_{y_i}, m_{z_i}$ 、慣性テンソル I_{x_i} ないし I_{z_i} は、第 i の関節の動作により移動しかつ第 $i+1$ の関節の動作によっては移動しない第 i の関節に連結されるリンクの質量、同じくリンクの重心位置、同じくリンクの慣性テンソルをそれぞれ示す。なお、慣性テンソルの演算手順は、技術文献 1 の節 3.8 に詳しく説明されているので（式(3.38)を参照）、ここでは説明を割愛する。

【0077】上記した重心位置および慣性テンソルを演算するうえでの基準は、リンク座標系に置かれる。関節ごとに割り当てるリンク座標系の定義手順が、吉川恒夫訳「ロボット・マニピュレータ」コロナ社、初版・1989年12月15日発行（以下技術文献 2 という）の節 2.9 に

詳細に説明されているが、ここでは、基本的なリンク座標系の定義を図 5 において説明する。例えば、第 2 関節 7_2 と第 3 関節 7_3 に跨るリンク L_2 においては、第 2 関節の回転中心を原点として第 2 関節に割り当られるリンク座標が定められ、その x 方向をリンク L_2 の長さ方向とし、z 方向を第 2 関節の回転方向、y 方向を右手座標系の方向と決める。

【0078】次に、慣性テンソル I は一般に物体の回転しにくさを示すために使用され、よく知られている限りの 3×3 の対称行列である式(2-2)で表される。その対角成分はロボット機構部の各関節の慣性モーメントを、その他の成分は慣性乗積の負の値を表すものとなっている。

【数9】

$$I = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \quad (2-2)$$

【0079】また、慣性モーメントは次式で求められ、物体(ロボットリンク)の回りにくさおよび止めにくさ

$$I_x = \int (y^2 + z^2) dm \quad (2-3)$$

$$I_y = \int (z^2 + x^2) dm \quad (2-4)$$

$$I_z = \int (x^2 + y^2) dm \quad (2-5)$$

【0080】次いで、慣性乗積は次式より求められ、物体(ロボットリンク)が回転運動を行っているとき、そ

$$I_{xy} = \int xy dm \quad (2-6)$$

$$I_{yz} = \int yz dm \quad (2-7)$$

$$I_{zx} = \int zx dm \quad (2-8)$$

ここで、dmは微小体積の質量を示す。また、式中のx, y, zは、所定のリンク座標系を基準として表され、図5に示すリンクL₂の慣性テンソルの基準座標は、第2関節に割り当てられたリンク座標系をリンクL₂の重心位置に移したときの座標系とする。他の関節についても同様に定義する。

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (2-9)$$

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}^T \quad (2-10)$$

$$G = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \quad (2-11)$$

【0082】式(2-9)ないし式(2-11)の各構成要素は表3に示されたパラメータならびに関節位置θ, 関節速度θ_dのみから構成され、θ, θ_d以外はすべて既知の値である。従って、ステップ2で求められる関節位置応答

$$\theta = \theta_r$$

$$\theta_d = \theta_{dr}$$

【0083】ステップ3でH, C, Gの各行列が算出されたことで、式(2-1)中の未知の値は関節加速度θ_{dd}ならびに重力加速度が働く方向を示す重力加速度ベクトルgの二つとなる。後者はロボットの設置姿勢によって変わるものであるが、一般的にはロボットの設置姿勢が変わらないと考えて差し支えなく、従って予め規定しておくことができる。

【0084】ステップ4において、最終的に式(2-1)で

を表す。

【数10】

の運動を乱すモーメントを表す。

【数11】

$$\theta = \theta_r \quad (3-1)$$

$$\theta_d = \theta_{dr} \quad (3-2)$$

【0081】なお、慣性、粘性および重力の行列値の演算方法は技術文献1に詳しく説明されている。ここではその説明は省くが、3つの関節からなるロボットの場合、慣性、粘性および重力の行列は次式のように表される。

【数12】

θr, 関節速度応答θ_{dr}rを次式のごとくθ, θ_dに置けば、式(2-1)中の慣性行列H, 粘性行列Cおよび重力行列Gを求めることができる。

【数13】

$$\theta = \theta_r \quad (3-1)$$

$$\theta_d = \theta_{dr} \quad (3-2)$$

表される逆動力学モデルを計算するうえで必要な値のうち未知なものは、関節加速度θ_{dd}だけとなる。これもステップ2で求められた関節加速度応答θ_{dd}rを式(4-1)のかたちで代入すれば、衝突のない場合にロボットの各関節がθr, θ_{dr}r, θ_{dd}rで運動するために必要となる各関節での発生トルクTを、式(4-2)で予測することができる。

【数14】

$$\theta_{dd} = \theta_{dd,I} \quad (4-1)$$

$$T = H(\theta_r) \cdot \theta_{dd,r} + C(\theta_r, \theta_{dr}) + G(\theta_r) \cdot g \quad (4-2)$$

$$T = [T_1, T_2, T_3]^T \quad (4-3)$$

【0085】ステップ4で算出した衝突が起こっていない場合にロボットの各関節が $\theta_r, \theta_{dr}, \theta_{dd,r}$ で運動するために必要な各関節のトルク T と、ステップ2で算出した実際にモータが関節を駆動しようとしているトルク

$$\Delta T = T_c - T \quad (5-1)$$

$$\Delta T = [\Delta T_1, \Delta T_2, \Delta T_3]^T \quad (5-2)$$

【0086】衝突がなければ、トルク指令値 T_c と予測値 T とはほとんど等しく、 $\Delta T \approx 0$ となるはずである。一方、 ΔT の絶対値が予め設定された判定値 ϵ_i よりも大きくなる場合には衝突が発生したものと見做して、制

$$|\Delta T_1| > \epsilon_1 \quad (5-3)$$

$$|\Delta T_2| > \epsilon_2 \quad (5-4)$$

$$|\Delta T_3| > \epsilon_3 \quad (5-5)$$

$$\epsilon = [\epsilon_1, \epsilon_2, \epsilon_3]^T \quad (5-6)$$

【表5】

記号	項目	単位	個数
ΔT	トルク予測値と実際の発生トルクとの差	N・m	$n (= 3) \times 1$ 行列
ϵ	衝突判定値	N・m	$n (= 3) \times 1$ 行列

【0087】ちなみに、 ϵ_i は例えば $0.1 T_{ci}$ もあれば十分であることが多い。現実には $0.3 T_{ci}$ なければならない場合もあるが、そのときでも3割増し以上のトルクが掛かれば衝突と判定することになるわけで、従来技術の項で述べた位置偏差による場合、トルクに換算すれば例えば $0.8 T_{ci}$ もしくはそれ以上にも及ぶのに比べると、本発明の場合衝突検出精度が格段に向上升ることが分かる。このような向上は、関節によってはさらなる改善も期待できる。

【0088】なお、判定値 ϵ_i は固定しておくといった単純な扱いをしてもよいが、加速度の大きさや速度の大きさに応じて可変とし、加速度や速度が大きければ感度を下げ、小さければ上げるといったように感度変更可能としておくこともできる。加減速時に感度を上げて $0.1 T_{ci}$ といったようにしておくと衝突の誤検出を招きやすく、加減速が少なくかつ低速のときに $0.3 T_{ci}$ といったように感度を下げておくと衝突の検出が遅れることになるからである。

【0089】以上の説明から分かるように、ロボットの衝突の検出のためには、全ての関節を駆動する各モータの情報を統括して処理し、それらを統合して逆動力学演算することにより全ての干渉（衝突）を見い出すことが

トルク指令値 T_c との差分 ΔT をステップ4において計算する。

【数15】

御プログラムを停止処理へと移行させる。具体的には、式(5-3)ないし式(5-5)の条件式のいずれか一つでも満足する状態が生じれば、衝突が発生したと判定される。

【数16】

$$|\Delta T_1| > \epsilon_1 \quad (5-3)$$

$$|\Delta T_2| > \epsilon_2 \quad (5-4)$$

$$|\Delta T_3| > \epsilon_3 \quad (5-5)$$

$$\epsilon = [\epsilon_1, \epsilon_2, \epsilon_3]^T \quad (5-6)$$

できるようにしている。その演算において、式(2-1)が不可欠であることは言うまでもない。ロボットの機構パラメータのうちリンク質量・リンク重心位置・リンク長さも欠くことのできない要素である。しかし、リンク慣性テンソルは細長いリンクを採用するロボットでは必要となるが、軽量ロボットまたは短リンクロボットなどの場合に考慮するに及ばないので、その都度適宜選択することにすればよい。

【0090】サーボドライバ $2_1, 2_2, 2_3$ に入力される「位置指令」はハードディスク15に記憶されたデータに基づき軌道計画部22で演算された結果として出された命令である（図1を参照）が、教示中にあってはティーチペンダント14（図3を参照）によって入力されたデータに基づく「位置指令」ということになる。それゆえ、教示中の不意の衝突に対しても、本発明の動作を行わせることができる。

【0091】本発明に係る衝突検出法によれば、衝突検出が精度高く迅速になされるので、後述する停止処理操作とあいまって、少なくともロボットが壊れたり周辺に存在する治具等の機械的構成要素を損傷させたり、復旧に多大の時間を要することは可及的に少なくなり、衝突後の移動や教示を速やかに再開することができるよう

なる。衝突検出はコントローラに搭載した衝突検出ソフトによるので、ショックセンサや機械式衝突検出装置をロボットリンクに装備する必要はなく、ロボットリンクの構造や機構の複雑化は回避され、また軽量化も図られる。

【0092】ところで、図1においては、サーボドライバからコントローラに、モータの位置応答、モータの速度応答、モータの加速度応答およびモータへのトルク指令値が伝達されると説明した。しかし、速度応答と加速度応答とはそれぞれの前者を微分すれば得られるものであるので、サーボドライバからコントローラへは、図6のようにモータの位置応答と速度応答にとどめ、衝突検出部11において速度応答値を微分してモータの加速度応答を得るようにもよい。さらには、モータの位置応答のみをコントローラへ供給し、図示しないが、コントローラで一階および二階微分して速度および加速度を求めるようにすることも可能となる。いずれの場合も、サーボIF20を介した情報伝達の混雑を緩和させることができると期待できる。

$$T_{ci'} = T_{ni} + \dot{\theta}_i$$

【0095】モータがブラシレスDCモータである場合には、実際に流れている全ての電流が有効なトルクになるわけではない。特にモータの高速回転時には、トルクに変換されない無効電流が増加し、指令と出力の誤差が大きくなる。実際にモータが出力しているトルクは、モータの三つ相のうち二つの相に流れている電流（以下相電流という）を検出して、図8に示す座標変換部24で座標変換を行うことにより、トルクに変換される電流である有効電流*i_{qa}*を求めることができる。

【0096】座標変換処理は無効電流を捨て有効分だけ

$$i_{qa} = \sqrt{2} \cdot \cos(\theta_{re} + \pi/3) \cdot i_{ua} + \sqrt{2} \cdot \cos(\theta_{re}) \cdot i_{va}$$
(6-1)

【0097】なお、電気角θ_{re}の検出には多くの方法があるが、絶対値エンコーダを使用する場合は次のようにして求めることができる。まず、電気角360°に相当するエンコーダのパルス変位θ_{e_rev}を算出する。θ_{e_rev}は、エンコーダ1回転当たりのパルス変位θ_{m_rev}をモータの極対数Pnで除した値であり、式(6-2)で算出される。極対数とはブラシレスDCモータのu, v, wの三つの極対がモータ1回転当たりに配置されている数であって、一つの極対に相当するモータの回転量が電気角360°となる。エンコーダ1回転当たりのパルス変位θ_{m_rev}、モータの極対数Pnについても、それぞれエンコーダとモータの仕様から予め規定しておくことができる値である。

【0093】上記した例では、差分△Tを求めるにあたり、逆動力学演算に基づくトルク予測値と、サーボドライバが速度制御により生成したトルク指令値Tmを実際の関節駆動トルクと扱い、そのトルク指令値を採用すると説明した。しかし、そのトルク指令値に代えて、各モータにおけるトルク応答値から算出されるトルクを採用してもよい。

【0094】図7から分かるように、サーボドライバ2₁, 2₂, 2₃で求められたトルク指令値T_{m_i}に応じた電力を動力線から各ブラシ付きDCモータ1₁, 1₂, 1₃に供給するサーボAMP 2a₁, 2a₂, 2a₃からの回路に電流検出器23₁, 23₂, 23₃を設ける。そして、電流検出器で拾った電流をA/D変換した後にそのトルク値T_{n_i}にトルク定数を乗じて、N·mとして得たトルク応答値（実際に出力されているトルク）T_{c_i'}を使用する。この場合、前記した式(1-3)の代わりに次式が採用されることになる。

【数17】

(1-13a)

取り出すというものであり、これについては、杉本英彦他著「ACサーボシステムの理論と設計の実際」総合電子出版社、第2版・1990年5月8日発行（以下技術文献3という）の節4.2で詳細に説明されている。最終的には当文献中の式(4.11)で表される。式中の*i_{qa}*の項が有効電流に相当するもので、相電流*i_{ua}*, *i_{va}*とモータの電気角θ_{re}より有効電流*i_{qa}*が式(6-1)で算出される。

【数18】

(6-1)

wの三つの極対がモータ1回転当たりに配置されている数であって、一つの極対に相当するモータの回転量が電気角360°となる。エンコーダ1回転当たりのパルス変位θ_{m_rev}、モータの極対数Pnについても、それぞれエンコーダとモータの仕様から予め規定しておくことができる値である。

【数19】

(6-2)

には、式(6-3)となる。式中の演算子MODは、除算の余りを算出することを表している。

【数20】

$$\theta_{re} = 2\pi \cdot \text{MOD}(\theta_e, \theta_{e_rev}) / \theta_{e_rev} \quad (6-3)$$

nの比で、モータの仕様から予め規定しておくことができる値である。

【数21】

$$T_n = K_t \cdot i_{qa}$$

【0100】上に現れた記号を参考までに、表6に掲記しておく。なお、上記した図8は出力トルクの検出機能を持つサーボドライバのブロック線図であり、電流検出器23_u, 23_vで検出された相電流*i_{ua}*, *i_{va}*が、A

(5-4)

D変換器を介して読みとられている。電気角の算出のため必要となるエンコーダの出力パルスも、読みとり可能となっている。

【表6】

記号	項目	単位	数
T _n	モータ出力トルク	[N・m]	1
i _{ua}	有効電流	A(アンペア)	1
θ _{ra}	モータの電気角	rad	1
i _{ua}	モータ u 相の相電流	A(アンペア)	1
i _{va}	モータ v 相の相電流	A(アンペア)	1
θ _{m,rev}	エンコーダ 1 回転あたりのパルス変位	P(パルス数)	1
P _n	モータの極対数	—	1
θ _e	エンコーダからの出力パルス	P(パルス数)	1
K _t	トルク定数	N・m/A	1

【0101】なお、モータがブラシ付きDCモータの場合には、図8中の座標変換は必要なく、直ちにトルク定数を乗じればよい。このようにモータの出力トルクを検出して、これを衝突検出に供するようにすれば、より一層検出精度が向上する。とりわけ、高速回転時にトルクに変換されない無効電流が増加するためにトルク指令値とモータ出力トルクとの間にずれの生じやすいブラシレスDCモータ等においては、極めて有効な手段となる。

【0102】図1では、サーボドライバから拾い上げた位置・速度・加速度応答値を用いて逆動力学演算するようしているが、拾い上げてから演算に入る関係上、後述する停止処理に入るまでに若干の遅れが出る。図9はそれを改善しようとするもので、衝突が発生していない場合に各関節を駆動するモータが発生すべきトルクを予測するときに使用されるモータの位置・速度・加速度として、作業データの再生動作のために演算した関節位置とその関節位置をもとに計算した速度および加速度を採用するものである。

【0103】モータがいかなる位置・速度・加速度で動いているかは軌道計画の段階で分かるはずであり、これを使用すればサーボの状態を見なくても予測トルクを計算しておくことができるとの知見に基づくものである。従って、軌道計画を解析した結果得られる位置・速度・加速度から各モータで発生すると思われるトルクを逆動力学演算によって求め、それを実際にモータに流れているトルクと突き合わせて衝突の有無を検出しようとするのである。

【0104】そのために、図9のコントローラ13内の制御プログラムには、図1に示した軌道計画部22, 補間点計算部16, 逆変換部17に加えて、補間周期ごとのロボットの各関節位置からロボット各関節の速度・加速度を計算する速度・加速度計算部25が付加されている。

【0105】なお、データ変換部18からサーボドライ

バ2への位置指令信号の出力と逆動力学演算のためのデータ信号26の出力との間に時間的調整をとっておく必要があるが、その場合、後者を前者より例えばサーボドライバでの演算時間だけ遅らせるといった配慮をすればよい。この図9のような構成にすれば、サーボドライバ2とコントローラ13との間のデータのやり取り量も少なくなり、通信の負担が軽減されることになる。ちなみに、ティーチペンダントによる教示中や手動運転中でも、軌道計画部22で作業軌道を生成して関節の位置・速度・加速度を予測することが可能であるので、この方法は先の例(図1を参照)と同様に、動作再生時以外でも適用することができる。

【0106】衝突の検出は、ロボットの作業軌道の補間点データに対してサーボドライバ2のフィードバック制御が行われている限り、図10に示す6つのステップを順次繰り返して行うことにより実現される。なお、衝突検出は、コントローラ13内のプログラムである衝突検出部11で行われることは、図1の例と異なるところがない。

【0107】まず、ステップ11においては、ロボット各関節の位置・速度・加速度が、ハードディスク15に記憶される作業データの再生動作行う過程で計算される。具体的には、速度・加速度計算部25で補間周期ごとのロボットの各関節位置からロボット各関節の速度θ_d、加速度θ_{dd}が計算される。

【0108】ここで、ロボット各関節の速度・加速度を計算する方法について説明する。補間周期S_{mpt}ごとのロボット各関節の位置θが算出されると、現周期で算出された関節位置をθ_{curr}とし、直前の補間における関節位置をθ_{prev}として、現補間点における各関節の速度θ_dが以下のようにして算出される。

【数22】

$$\theta_d = (\theta_{curr} - \theta_{prev}) / S_{mpt}$$

(7-1)

【0109】現周期で算出された関節速度を $\theta_d curr$ とし、直前の補間時の関節速度を $\theta_d prev$ とすれば、現補間点における各関節の加速度 θ_{dd} が同様の要領で求められ

$$\theta_{dd} = (\theta_d curr - \theta_d prev)$$

【0110】ステップ11で算出したロボット各関節の位置 θ から、サーボドライバへの位置指令 θ_c を式(8-1)により算出し、サーボIF20を介してサーボドライバ2へ出力する。モータは減速機を介して関節を駆動するので、ロボット各関節の位置 θ に減速比を乗じた値

$$\theta_c = [\theta_{c1}, \theta_{c2}, \theta_{c3}]^T \quad (8-1)$$

$$\theta_{ci} = \theta_i \cdot c_i \quad (8-2)$$

【0111】ステップ13ではトルクの状態の読み込みがなされる。すなわち、衝突検出部11がサーボIFを介してサーボドライバ2から、ロボットの各関節を駆動するモータのトルク指令値 T_{m1} , T_{m2} , T_{m3} を読み込む。また、モータからロボット関節への結合は減速

$$T_c = [T_{c1}, T_{c2}, T_{c3}]^T \quad (9-1)$$

$$T_{ci} = T_{mi} \cdot c_i \quad (9-2)$$

【0112】以後、ステップ14では逆動力学モデル(H, C, G)が計算され、ステップ15で逆動力学演算によるトルク予測値が演算される。ステップ16で前述したとおりの手順によって衝突判定処理がなされる。もちろん、その後には後述する停止処理も行われる。

【0113】図9の例においては位置・速度・加速度指令をもとにトルクを予測しており、実際の応答値と指令値の誤差が小さければトルク予測の精度が上がることは明らかである。従って、衝突検出精度も向上することになる。そこで、図9の構成に、サーボドライバ2内のサーボ制御系に対して速度指令をフィードフォワード補償すれば、サーボドライバの指令追従性能を上げることができ、常にロボットの衝突検出精度を高く維持しておくことができるようになる。図11は、図9の構成に速度指令のフィードフォワード補償を付加した形態を示している。

【0114】もう少し詳しく述べると、図9の場合、軌道計画の後に関節速度も演算されているので、図11のように速度をデータ変換部27からフィードフォワードすることができる。フィードフォワードすれば、位置・速度・加速度の関係が実際の応答と異なる程度を図9の場合よりも小さくすることができる。すなわち、図9の場合、位置・速度・加速度と実際に使用される位置・速度・加速度とが乖離する傾向にあり、従って衝突していないにもかかわらず逆動力学演算結果とトルク指令値とが一致しなくなるといった演算精度上の問題が生じる。しかし、フィードフォワード補償によれば、これを効果的に解決することができる。

る。なお、時刻0における、各関節の速度、加速度はいずれも0と置かれる。

【数23】

$$Smp t \quad (7-2)$$

がサーボドライバへの指令値となる。このようにして、サーボドライバへの位置指令値の送信がステップ12で行われる。なお、 θ_c はロボット関節の位置応答 $[r_a d]$ であり、その個数はn (=3) × 1行列となる。

【数24】

$$(8-1)$$

機を介して行われるので、サーボドライバ内のトルク指令値 T_{mi} に減速比を乗じた値が関節を駆動するトルク T_{ci} として与えられる。

【数25】

$$(9-1)$$

$$(9-2)$$

【0115】図12には、速度指令のフィードフォワード補償の有無によって速度応答がどうのよう相違するかが示されている。速度指令のフィードフォワード補償を行えば、(a)と(b)を見比べて分かるように、破線で示す速度指令に対して実線で示すごとく追従性能(速度応答)が向上する。これから分かるように、フィードフォワードすればタイムラグが小さくなり、結果として、位置・速度・加速度の関係も実際に極めて近いものとなる。

【0116】なお、フィードフォワード補償すると位置フィードバック比例ゲイン K_{pp} の効きは落ちるが、速度はフィードフォワード的に効くので速度指令が迅速になれる。ただ速度だけでは十分な位置精度が出ないので位置をもとにした速度制御も残されており、これを速度のフィードフォワードに対して補間的に機能させておくことになる。

【0117】ちなみに、式(2-1)は $J^T K$ を省いたものであると説明したのは、常に外部の何物にも触れずに作業した空間を変位する例えばアーク溶接ロボットに適用する場合を念頭に置いたからである。そこで、本発明をワークに対して接触しながら作業する類のロボットに適用する場合は、ツールを現在位置から次の目標位置まで単に空間移動させるといったときのみ有効となる。しかし、式(2-1)に $J^T K$ を付加しておくなれば、逆動力学演算に多少の処理時間増加を伴うことになるが、力覚センサ等を働かせながらワークに対して接触する作業をしているときも本発明の思想を反映させることができるようになる。

【0118】以上までは衝突の検出方法を述べたが、これからは衝突の検出があった後の停止処理について説明する。その処理の趣旨は、以下のとおりである。衝突が発生したときは、いち早くそれを検出すべきであることは言うまでもない。しかし、衝突を検出できた時点では衝突してから既に少し時間が経過しているわけであるから、衝突直前にロボットを停止させるということは至難の技である。

【0119】そこで、ロボットの停止操作においては、衝突してもロボット本体やぶつかった対象物や障害物の損傷が最小限に抑えられるようにすることを最終的な目標とすべきである。簡単に言えば、各モータには、その関節を介して動かされているリンクが今持っている慣性モーメントを吸収して停止させることができる程度のトルクしか発生できないようにしておき、これによって衝突逃げ動作を可能にして被害を最小限に止めるとの観点に立った停止処理を行うこととする。

【0120】この停止処理は図2のところでも触れたように、停止もしくは低速で動いている関節については、矢印9のような衝突による干渉力を和らげる方向への回転を許容できる制御をすることであり、従来技術の項で述べたように、どの関節においても可能な限り大きいトルクを発生させて迅速に制動しようとする場合、ロボットと障害物との間に大きな力が作用することになるというのを回避しようとするものである。

【0121】具体的には、衝突検出できた以降は、各モータ速度を0とするために必要な最小限のトルクを越えるトルクを発生させないように、モータの発生トルクを制限することにする。例えば第3関節は止まっておりしかも重力も作用していないとするとトルクはほとんど0でよいから、ぶつかればその関節は簡単に回転し、リンクは障害物からスルッと逃げる。負荷を伴うことなく逃げることができれば、第2関節や第3関節には停止するための情走の余地も残され、ロボット全体の干渉力や衝撃力を和らげ被害の発生が最小限に抑えられる。このように、本発明においては、大きな慣性力を持っていないかまたは高い速度が発生していない関節については、力に逆らわずに逃げる仕組みを提案しようとするものである。

【0122】その停止処理の主旨は、関節のうち停止もしくは低速で動作している関節あるいは慣性力が小さいため動作停止に必要なトルクが最大トルクと比較して十分小さい関節については、衝突があった後に障害物に対して逃げ動作をさせ、衝突による干渉力が和らげられるようにしたことである。

【0123】また、衝突発生時の各関節の速度と負荷イナーシャからロボットの動作を停止させるために必要な最小限のトルク値を関節ごとに算出し、その最小限のトルク値をサーボドライバ内のトルク制限器3に設定すると共に、全てのサーボドライバ内の位置フィードバック

比例ゲインKppを0に置き替えるようにする。それによって、強制的に速度指令値を0とした速度フィードバック制御による減速処理を行わせるのである。

【0124】さらに、ロボットの動作を停止させるために必要な各関節のトルク値の算出においては、全ての関節が同じ時間で停止すると仮定し、関節ごとに減速停止時間を算出した後、全ての減速停止時間のうちの最大値をいずれの関節の減速停止時間にも充てることにより、上記した逃げ動作を幾つかの関節において実現できるようにするものである。

【0125】衝突が検出されても速度はステップ状に0となることがないので、衝突発生時の関節速度θ_d、rはサーボドライバの状態変数であるモータ速度θ_d、mから知ることができる。関節位置についても、サーボドライバの状態変数であるモータ位置θ_mから知ることができる。衝突発生時の関節位置が分かれれば、負荷イナーシャであるH行列の計算は可能となる。これは当然ロボットの姿勢によって変化するイナーシャである。この衝突発生時のイナーシャと速度から以後速度を0にするに必要なトルクを計算し、この値以上のトルクが発生しないようにモータ出力トルクを制限する。このトルク出力制限処理は、図1、図9、図11に表されたトルク制限値の伝達経路28を介して行われる。

【0126】図2のように第3関節7₃は動いていないが第1関節7₁と第2関節7₂が動いているとすると、第1関節と第2関節についてはその慣性力を制御するだけの比較的大きなトルクが必要となるが、第3関節では慣性力は小さく従って減速に必要なトルクも第1関節や第2関節に比較すれば小さくてよい。そのため、第3関節7₃を駆動するモータのトルク制限器には小さい値が設定され、第3関節に連結されたリンクが第1関節、第2関節の動きによって衝突しても、第3関節は簡単に衝突による干渉力を和らげる方向へ逃げることが可能となり、衝突による変化を吸収することができる。

【0127】以下に、停止処理の具体例を述べる。前述したいづれかの衝突検出処理により衝突を検出した後、図13の4つのステップからなる処理を行う。まず、ステップ21において、ロボットの動作停止に使用可能なトルク値を算出する。

【0128】衝突発生時の停止処理に使用可能な各関節のトルク値Tusabが、式(4-2)を変形した式(10-1)から算出することができる。このトルク値Tusabは、動作を停止させるための減速運動に使用できるもので、加速度によって生じるトルクと見做される。重力によるトルクや粘性によるトルクは速度に依存するものであり、加速度によるトルクとは別に確保されねばならないトルクである。

【0129】式(4-2)を変形して速度により消失するトルク(粘性トルク)と重力により消失するトルクとを引き去れば、それが減速に供し得るトルクとなる。すなわ

ち、遠心力や重力加速度に影響する外乱トルクによつて、実際に停止のための減速運動に使用できるトルクが

制限される。

$$T_{usab} = -\text{sign}(\theta_{dr}) \cdot T_{max} - C(\theta_r, \theta_{dr})$$

$$- G(\theta_r) \cdot g \quad (10-1)$$

$$T_{max} = [T_{max1}, T_{max2}, T_{max3}]^T \quad (10-2)$$

$$T_{usab} = [T_{usab1}, T_{usab2}, T_{usab3}]^T \quad (10-3)$$

$$\begin{array}{ll} \text{ただし, } \theta_{dr} \geq 0 \text{ のとき} & \text{sign}(\theta_{dr}) = 1 \\ \theta_{dr} < 0 \text{ のとき} & \text{sign}(\theta_{dr}) = -1 \end{array}$$

【0130】なお、式(10-1)中の $\text{sign}(\theta_{dr})$ は符号を決めている。 T_{max} は各モータが発生し得る最大トルクであって、正の値に設定される。しかし、場合によってはマイナス値として使用したいときもあるので、 $\text{sign}(\theta_{dr})$ はそれを判定している。現在の速度が正であれば負の加速度が必要となり、負のトルクが与えられる。正の速度が出ていると T_{max} は正となるが、負の符号がついているので負のトルクとなる。負の速度が出ていれば正の加速度が必要となるので、負×負=正となるようにしている。すなわち、速度(θ_{dr})に対して符号を反転させることを意味している。

【0131】 $-\text{sign}(\theta_{dr}) \cdot T_{max}$ は、各関節が出力できるトルクの最大の減速トルクである。現在の速度を0にするために必要な減速トルクは、そのトルクから

$$T_{maxi} = T_{mmaxi} \cdot \zeta_i$$

【0133】 T_{mmaxi} も減速比 ζ_i も既知の値であるから T_{maxi} も予め固定的に与えておくことができる。C行列やG行列も衝突を検出する過程で計算されており、新たに計算する必要はない。現時点のものでよいから、式(4-2)で使ったものをそのまま持つてくればよい。本当は時間のずれが極く僅か存在するが、衝突した

速度によって消失するトルクと重力によって消失するトルクを引算したもので、それが実際に減速のために使うことができる式(10-1)で与えられるトルク T_{usab} である。

【0132】式(10-1)は、式(4-2)の $H(\theta_r) \cdot \theta_{dd}$ を T_{usab} と、式(4-2)のTを $-\text{sign}(\theta_{dr}) \cdot T_{max}$ と置き換えたものである。なお、第*i*関節の最大出力トルク値 T_{maxi} は、関節を駆動するモータが出力可能な最大トルクを T_{mmax} とするとき、減速比を考慮して式(10-4)で算出することができる。言うまでもなく、 T_{mmax} はモータの仕様から予め規定できる値であり、粘性行列Cならびに重力行列Gは、衝突発生時の各関節の位置と速度から算出可能である。

【数27】

$$(10-4)$$

時から再度計算するには及ばないからである。もちろん、計算してもよいが時間を要するうえに計算値に大した違いは生じない。なお、上に現れた記号を参考までに、表7に掲記しておく。

【表7】

記号	項目	単位	個数
T_{max}	関節が出力可能な最大トルク	N・m	$n (= 3) \times 1$ 行列
T_{usab}	動作停止に使用できるトルク	N・m	$n (= 3) \times 1$ 行列
T_{mmax}	モータが出力可能な最大トルク	N・m	$n (= 3) \times 1$ 行列

【0134】ステップ22では、動作を停止させるために必要な各関節のトルク値が算出される。減速により発生するトルクは、慣性行列Hと各関節の減速度の積であり、式(11-1)で表される。なお、 θ_{dd} は未知であるが、

$$H(\theta) \cdot \theta_{dd} = T_{dec}$$

【0135】ところで、この停止処理の制御においては、全ての関節が時間 t_{md} で速度0に減速停止するものと仮定する。すると、各関節の減速度 θ_{dd} は、式(11-2)で表現することができる。なお、式(11-2)中の θ_{dr}

$$\theta_{dd} = \theta_{dr} / t_{md}$$

【0136】全ての関節が式(11-2)の時間 t_{md} で減速停止することを制御条件としているのは、何らかの条件を課さなければ制御の実行が図られないためであるが、この条件によって、幾つかの関節において上記した逃げ動

分かっていればこの式によって T_{dec} が求まることになる。

【数28】

$$(11-1)$$

は、衝突発生時の各関節速度である。ただ、いまの段階では t_{md} は未知である。

【数29】

$$(11-2)$$

作を可能ならしめることができるようになるのである。

【0137】そこで、時間 t_{md} として、後で説明する式(11-4)ないし式(11-6)でロボットの関節の数だけ求まる複数の t_{md} の中から、最長のものを使用することにする

(式(11-7)を参照)。これは、ロボットの全ての関節において衝突後のロボットの減速運動により発生するトルクが、減速に使用可能なトルク T_{usab} を越えないようにしておいためであり、それを実現する t_{md} によって、全ての関節を同じ時間で停止させることができるようになる。

【0138】なお、参考までに記せば、選定された t_{md} の関節は、結果的に最も早く動いている関節あるいは一番大きいモーメントを受ける関節であることが多い。ちなみに、このようにして選定された時間によれば、最も遅く動いている関節では速度が0になるまでに時間が十

$$II(\theta_r) \cdot \theta_{dr} / t_{md} = T_{usab}$$

【0140】この式(11-3)から、停止処理の減速に使用できるトルク T_{usab} の範囲内で、ロボットの各関節の速度を0にできる減速停止時間 t_{md} が算出できる。なぜなら、式(11-3)において、 $H(\theta_r)$ は、衝突発生時の各関節位置 θ_r から算出できかつ θ_r はサーボIF20を介してモニタ可能であるからである。また、 θ_{dr} は衝突発生時の各関節速度でありサーボIFを介してモニタ可能であり、 T_{usab} は式(10-1)で計算済みでもある。

【0141】最終的には未知の値は t_{md} だけとなる。 t_{md} は行列式であり、3つの関節からなるロボットの場合

$$\begin{aligned} t_{md1} &= H_{11} \cdot \theta_{dr1} / T_{usab1} + H_{12} \cdot \theta_{dr2} / T_{usab1} \\ &\quad + H_{13} \cdot \theta_{dr3} / T_{usab1} \end{aligned} \quad (11-4)$$

$$\begin{aligned} t_{md2} &= H_{21} \cdot \theta_{dr1} / T_{usab2} + H_{22} \cdot \theta_{dr2} / T_{usab2} \\ &\quad + H_{23} \cdot \theta_{dr3} / T_{usab2} \end{aligned} \quad (11-5)$$

$$\begin{aligned} t_{md3} &= H_{31} \cdot \theta_{dr1} / T_{usab3} + H_{32} \cdot \theta_{dr2} / T_{usab3} \\ &\quad + H_{33} \cdot \theta_{dr3} / T_{usab3} \end{aligned} \quad (11-6)$$

$$t_{md} = \text{MAX}(t_{md1}, t_{md2}, t_{md3}) \quad (11-7)$$

【0142】動作を停止させるために必要な各関節のトルク値 T_{dec_i} は、式(11-4)ないし式(11-7)から算出される減速停止時間 t_{md} を用いて、式(11-8)で算出される。なお、 T_{dec_i} は停止に必要な関節のトルク値 [N

$$T_{dec_i} = H(\theta_r) \cdot \theta_{dr} / t_{md}$$

【0143】このようにして算出された T_{dec} は、衝突発生時のロボットの各関節の位置・速度により変化する負荷イナーシャや外乱トルクの変動を考慮したもので、停止時間に遅延を生じさせない必要最小限のトルク値となる。

【0144】式(11-1)の H 行列、式(11-2)の現在の関節速度 θ_{dr} および減速停止時間 t_{md} から、各関節に必要な減速トルク T_{dec} が式(11-8)により分かる。どこか一つの関節すなわち減速動作に最大時間を要すると思われる関節については T_{usab} に等しくなるが、他の関節では $T_{dec} < T_{usab}$ となる。これは速度が小さいか慣性力

分にあり無用な惰走が強られることになるが、元来速度は低いので多少惰走してもその距離がさして大きくなることはない。

【0139】ここで、停止処理の減速に使用できるトルクは、式(10-1)から T_{usab} と算出されているので、式(11-1)の右辺を T_{usab} に置き換え、全ての関節が式(11-2)の時間 t_{md} で減速停止するものとする。式(11-1)中の θ_{dd} に式(11-2)の右辺を代入すると、式(11-1)は式(11-3)のように変形される。式(11-3)中の慣性行列 H の算出方法は、既に説明したとおりである。

【数30】

$$(11-3)$$

には、3つの関係式ができる。実際には、式(11-3)を、関節ごとに以下の3つの式に分解し、減速停止時間 t_{md} を算出する。具体的に言えば本実施例では関節数 $n=3$ であるから、関節ごとに減速停止時間 t_{md1} ないし t_{md3} を算出した後に、 t_{md1} 、 t_{md2} 、 t_{md3} の中の最大値を全ての関節の減速停止時間に充てる。式(11-7)中の MAX は要素中の最大値を抽出することを意味している。

【数31】

$$\begin{aligned} t_{md1} &= H_{11} \cdot \theta_{dr1} / T_{usab1} + H_{12} \cdot \theta_{dr2} / T_{usab1} \\ &\quad + H_{13} \cdot \theta_{dr3} / T_{usab1} \\ t_{md2} &= H_{21} \cdot \theta_{dr1} / T_{usab2} + H_{22} \cdot \theta_{dr2} / T_{usab2} \\ &\quad + H_{23} \cdot \theta_{dr3} / T_{usab2} \\ t_{md3} &= H_{31} \cdot \theta_{dr1} / T_{usab3} + H_{32} \cdot \theta_{dr2} / T_{usab3} \\ &\quad + H_{33} \cdot \theta_{dr3} / T_{usab3} \end{aligned} \quad (11-4)$$

$\cdot m]$ で、その個数は $n (=3) \times 1$ 行列である。 t_{md} は全ての関節での減速停止時間 [sec] で、個数は1である。

【数32】

$$(11-5)$$

$(=H)$ が小さいものについては T_{dec} も小さくなるからで、これがトルク制限値として設定されると、干渉力に対して逃げ動作を可能にする。

【0145】ステップ23では、サーボドライバ2内のトルク制限器3の設定値が変更される。すなわち、ステップ2で算出した T_{dec} は関節で発生するトルクであり、実際にモータで発生するトルクの制限処理を行うのはサーボドライバで行われる。通常、モータは減速機を介して関節を駆動しているので、 T_{dec} をモータシャフトの換算値 T_{m_dec} に変更し、サーボIF20を介した伝達経路28を通じて T_{m_dec} が各サーボドライバのト

トルク制限器3に設定される。

【0146】第*i*関節を駆動するモータにおける停止時間に遅延を生じさせない必要最小限のトルク値T_{m dec}_iは式(12-1)のようにして算出することができる。な

$$T_{m dec_i} = T_{dec_i} / \zeta_i \quad (12-1)$$

$$T_{m dec} = [T_{m dec_1} \cdot T_{m dec_2} \cdot T_{m dec_3}]^{\frac{1}{n}} \quad (12-2)$$

【0147】ちなみに、トルク制限値として式(11-8)のT_{dec}がサーボドライバに入ってくるが、T_{dec}はそのまま入れられない。動力学演算というは一般的に関節レベルで得られるので、求まったT_{dec}も関節レベルのトルクであって、実際にトルクを制限するのはモータである。モータと関節との間には減速機が介在されているから減速比で割った値を設定しなければならない。従つて実際はT_{m dec}_iが入れられることになる。

【0148】念のために記載するが、H行列やθ_d rが関節ごとに違うから、t_{md1} = t_{md2} = t_{md3}としているとはいえ、T_{m dec}_iはサーボドライバ2₁, 2₂, 2₃ごとに異なる。なお、得られたT_{m dec}_iをそのまま使用してもよいが、T_{m dec}_iに対して関節ごとにウェイトづけすれば都合のよい場合がある。そのような場合には、異なった係数を各T_{m dec}_iに乗じて採用すればよい。

【0149】ステップ24では、各サーボドライバ2内の位置フィードバック比例ゲインK_{pp}が、サーボIF20を介して0に変更される。以上のステップ21ないし24の処理により、強制的に速度指令値を0とした速度フィードバック制御による減速処理が行われる。

【0150】各サーボドライバ内のトルク制限器3には、停止に必要な最小限のトルク値が設定されるので、衝突による干渉力がこれを上回る場合、関節が干渉する力（衝突力）を逃す方向に動作し衝撃が軽減される。トルク制限器に最大トルク値が設定される従来の場合に比べれば、干渉力緩和効果が飛躍的に改善されることになる。

【0151】このように、トルク制限値を置き替えるだけでなくK_{pp}を0にしているのは、トルク制限値の置き替えだけでは関節が止まらないからである。位置指令や位置のフィードバックに如何なる値が来ていようとも、K_{pp}を0にすれば速度指令は直ちに0となる。速度指令を単純に0とするには、K_{pp}を0にすることが手っ取り早いからである。

【0152】上記から分かるように、速度ループだけでは関節の動きを止めにいこうとするのであるが、トルクが適当に変更して制限されているので、速く動いている関節では最大トルクで止める制御がなされる。一方、他の関節では、自己の速度を0に保持するだけのすなわち現在の姿勢を維持するだけのトルクしか発生しないようにしておくことができる。

【0153】これから分かるように、上記の停止処理に

お、T_{m dec}は停止に必要なモータのトルク値[N·m]であり、その個数はn (=3) × 1行列となる。

【数33】

$$T_{m dec_i} = T_{dec_i} / \zeta_i \quad (12-1)$$

$$T_{m dec} = [T_{m dec_1} \cdot T_{m dec_2} \cdot T_{m dec_3}]^{\frac{1}{n}} \quad (12-2)$$

よれば、トルク制限値を越える干渉力や衝撃が作用した場合に、その力の優劣を利用して関節に力逃しの挙動をさせることができ、ぶつかったことによる損傷の発生が可及的に抑えられる。なお、停止した状態でも、障害物にぶつかった関節であれぶつかっていない関節であれ速度ループにおける積分項はいつまでも機能するので、各リンクは関節が発揮している制限値内のトルクによって姿勢を維持し続ける。

【0154】ちなみに、例えば水平面内のみで変位するリンクであって、姿勢保持に重力に對抗する力を必要としないロボットの場合やトルクが掛け放しでは不味い境遇にあるリンクなどでは、速度が0になった時点でトルクを解放すればよい。その場合にはK_{vi}の入っている積分項の働きをなくせばよく、K_{pp}を0にするときもしくはその後に積分項をクリアする操作を附加しておけばよい。

【0155】以上で、衝突検出後の停止処理の説明を終えて、以下に、衝突検出精度の向上策について述べる。既に述べたように、トルク指令値に代えてトルク応答値を採用することもその一つであるが、ここではモータロータの慣性モーメントを考慮したり、モータ軸受や減速機の摩擦抵抗を考慮することによる衝突検出精度の向上について触れる。いずれも単独もしくは重複して機能させることができるものである。

【0156】まず、モータロータの慣性モーメントの考慮による衝突検出精度の向上から説明する。これは、前記したn × nの慣性行列における対角要素としての「関節の持つ加速度が該関節自体に及ぼすトルクの関係を示した慣性モーメント」、すなわち「関節位置により変動する負荷イナーシヤ」に、減速機を含むモータロータの慣性モーメントを減速比の2乗倍して加算しようとするものである。これによって、各関節内における減速機を含むモータロータの慣性モーメントにより発生するトルクも考慮して、衝突検出精度を向上させることができる。

【0157】まず、式(2-9)の慣性行列Hを以下のように変形することにより、各関節内における減速機を含むモータロータの慣性モーメントにより発生するトルクを考慮に入れる。ここで、I_mはモータロータの慣性モーメント [kg·m²]で、その個数はn (=3) × 1行列となる。

【数34】

$$H = \begin{bmatrix} H_{11} + I m_1 \cdot \epsilon_1^2 & H_{12} & H_{13} \\ H_{21} & H_{22} + I m_2 \cdot \epsilon_2^2 & H_{23} \\ H_{31} & H_{32} & H_{33} + I m_3 \cdot \epsilon_3^2 \end{bmatrix} \quad (13-1)$$

$$I m = [I m_1, I m_2, I m_3]^T \quad (13-2)$$

【0158】図1や図9さらには図11の例に共通して言えることであるが、慣性行列Hは各関節に連結されたリンクのイナーシャだけを計算しているに過ぎない。関節はモータで駆動されるから動くのであるが、関節を駆動する以前にモータも駆動されており、そのときのモータ自体の回転の慣性にも注目すべきである。

【0159】ここで、 $I m_i$ は減速機を含むモータロータの慣性モーメントである。連動して動く回転体の慣性モーメントは、その減速比の2乗分の1倍されて入力軸の慣性モーメントに影響を与える。従って、入力軸の慣性モーメントが出力軸に与える影響はその逆であるため、減速比の2乗倍されなければならない。減速機を含むモータロータの慣性モーメントが減速比の2乗倍されて加算されているのはそのためである。

【0160】慣性行列Hの対角要素はそれ自体に与える

$$T = H(\theta) \cdot \theta_{dd} + C(\theta, \theta_d) + G(\theta) \cdot g + \begin{bmatrix} C_{m1} \cdot \theta_{dm1} + |F_{cou1}| \cdot \text{sgn}(\theta_{dm1}) \\ C_{m2} \cdot \theta_{dm2} + |F_{cou2}| \cdot \text{sgn}(\theta_{dm2}) \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ C_{m3} \cdot \theta_{dm3} + |F_{cou3}| \cdot \text{sgn}(\theta_{dm3}) \end{bmatrix} \quad (14-1)$$

$$C_m = [C_{m1}, C_{m2}, C_{m3}]^T \quad (14-2)$$

$$F_{cou} = [F_{cou1}, F_{cou2}, F_{cou3}]^T \quad (14-3)$$

【表8】

記号	項目	単位	個数
C_m	減速機の動粘性係数	N・m・sec	n (= 3) × 1 行列
F_cou	クーロン摩擦トルク	N・m	n (= 3) × 1 行列

【0162】減速機自体にも摩擦がかなりあり、これも無視し得ない。速度項のことを粘性行列というが、それが $C_m \cdot \theta_d$ である。しかし、これは物が擦れるときの摩擦を対象とするものではない。或る速度で動いているものが或る速度で動いているものに重なると、その速度と速度が干渉しあって別方向にトルクが生じるが、そのトルクを意味する。これは、式(14-1)の4項目に表されている。

【0163】 C_m は動粘性係数であり、これに速度を乗じれば、外乱トルクとなる。 $|F_{cou}|$ はクーロン摩擦（静摩擦：動き始めるまでの抵抗）と言われるもので、速度に依存することなく一定の大きさを持つが、常に運

トルクの影響を示し、他の要素は他の関節に与える干渉トルクの影響を示す。減速機を含むモータロータの慣性モーメントが他の関節と干渉することはないので、対角要素のみを修正すればよい。なお、 $I m_i$ も ϵ_i^2 も予め設定可能な既知であるから、式(13-1)による場合も、既に述べた例と同様に衝突検出ならびに停止処理は可能である。

【0161】次に、摩擦抵抗を考慮して衝突検出の精度を上げることについて説明する。式(2-1)で表されるロボットダイナミクスモデルを、式(14-1)のように変更することにより、各関節内のモータの軸受や減速機の摩擦抵抗を考慮できるようにしようとするものである。なお、ここで現れる記号を参考までに、表8に掲記していく。

【数35】

$T = H(\theta) \cdot \theta_{dd} + C(\theta, \theta_d) + G(\theta) \cdot g + \begin{bmatrix} C_{m1} \cdot \theta_{dm1} + |F_{cou1}| \cdot \text{sgn}(\theta_{dm1}) \\ C_{m2} \cdot \theta_{dm2} + |F_{cou2}| \cdot \text{sgn}(\theta_{dm2}) \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ C_{m3} \cdot \theta_{dm3} + |F_{cou3}| \cdot \text{sgn}(\theta_{dm3}) \end{bmatrix}$

(14-1)

$$C_m = [C_{m1}, C_{m2}, C_{m3}]^T \quad (14-2)$$

$$F_{cou} = [F_{cou1}, F_{cou2}, F_{cou3}]^T \quad (14-3)$$

動と逆の方向に働くトルク成分である。 $C_m \cdot \theta_d$ m と $|F_{cou}|$ との和が、ここで考慮すべき摩擦トルクである。式(14-1)は式(2-1)に4項目を追加したものであり、摩擦を考慮した演算で精度を上げようとするものである。

【0164】ここで、 C_m はモータの軸受けや減速機の動粘性係数 $n \times 1$ 行列であり、動粘性係数とは、速度と動摩擦トルクとの比例関係を表す定数である。また、動摩擦トルクとは、速度に比例した大きさを持つトルク成分である。さらに、 θ_d m はモータの速度 $n \times 1$ 行列である。なお、式(14-1)中の $\text{sgn}(\theta_d m)$ は、モータの速度に基づいて、次式のように決定される符号関数であ

る。

$$\begin{aligned}
 \text{sgn}(\theta_{dm}) &= 1 & ; \theta_{dm} > 0 \text{ のとき} \\
 &= 0 & ; \theta_{dm} = 0 \text{ のとき} \\
 &= -1 & ; \theta_{dm} < 0 \text{ のとき}
 \end{aligned} \tag{14-4}$$

【0165】また、関節速度とモータ速度の関係はその減速比から、式(14-5)のように表される。

$$\theta_{dm} = \theta_d \cdot \varepsilon$$

ここで、 ε は関節内の減速機の減速比である。パラメータ C_m , F_{cou} , ε も予め設定可能な既知の値であるから、上記(14-1)によっても衝突検出ならびに停止処理できることは述べるまでもない。

【0166】

【発明の効果】本発明によれば、サーボドライバが検出したモータの位置・速度・加速度から衝突が発生していない場合に各モータが発生すべきトルクを予測し、そのトルク予測値と実際の関節駆動トルクとの差分を求め、その絶対値が予め設定された判定値より大きい場合に衝突が発生したものと見做すようにしたので、ロボットのいずれかのリンクやツールがワーク等の対象物やその他の障害物に衝突したことをトルクを通じていち早くかつ正確に検出することができるようになる。

【0167】衝突が生じていない場合に各関節を駆動するモータが発生すべきトルクを、ロボットの各関節の位置・速度・加速度とロボットの機構パラメータとをもとにして、 $T = H(\theta) \cdot \theta_{dd} + C(\theta, \theta_d) + G(\theta) \cdot g$ の逆運動力学演算によって予測するようにしておるので、そのトルク予測値を高精度に求めておくことができ、衝突センサ等の機械的構成物を搭載しなくても、迅速かつ正確に衝突を検知できる。

【0168】トルクを予測する際に使用するモータの位置・速度・加速度として、サーボドライバに位置指令すべく作業データの再生動作において演算した関節位置と、その関節位置をもとに計算された速度および加速度をあてがうようにすれば、サーボドライバからモータの位置・速度・加速度の情報を取り込む時間等だけ迅速に衝突検出演算を行わせることができる。

【0169】逆運動力学演算においては、ロボットの各関節の加速度により各関節で発生するトルクならびに他の関節の加速度により発生する干渉トルクを示す $n \times n$ の慣性行列、達心力とコリオリ力の影響を示す $n \times 1$ の粘性行列、および重力加速度の影響により発生するトルクを示す $n \times 3$ の重力行列を含むことしているので、その演算精度は極めて高いものとなる。

【0170】 $n \times n$ の慣性行列における対角要素としての「関節の持つ加速度が該関節自体に及ぼすトルクの関係を示した慣性モーメント」に、減速機を含むモータロータの慣性モーメントを減速比の2乗倍して加算しておけば、各関節内における減速機を含むモータロータの慣

【数36】

【数37】

$$(14-5)$$

性モーメントにより発生するトルクを考慮した衝突検出を行うことができる。

【0171】逆運動力学演算に供される式に、減速機の動粘性係数とクーロン摩擦トルクの要素を反映させるようにしておくと、各関節内のモータ軸受や減速機の摩擦抵抗を考慮した精度の高い衝突検出が実現される。

【0172】逆運動力学演算に供されるロボットの機構パラメータとして、関節に連結されるリンクの重心位置・質量・長さを与えておけば、演算が個々のロボットに則したものとなり、ましてや、慣性テンソルも付加しておけば、より一層精度のよいトルク予測が可能となる。

【0173】トルク予測値との差分が求められる実際の関節駆動トルクとして実際にサーボドライバが生成したトルク指令値から算出されるトルクを使用すれば、演算速度を高く維持することができる。各モータにおけるトルク応答値から算出されるトルクとする場合には、現実のトルクとの比較になって衝突検出のための精度がさらに向上する。

【0174】関節のうち停止もしくは低速で動作している関節あるいは慣性力が小さいため動作停止に必要なトルクが最大トルクと比較して十分小さい関節には、衝突があった後に逃げ動作させるようにしておけば、衝突による干渉力を軽減して物損の発生を抑制することができる。

【0175】衝突発生時の各関節の速度と負荷イナーシヤからロボットの動作を停止させるために必要な最小限のトルク値を関節ごとに算出し、その最小限のトルク値をサーボドライバ内のトルク制限器に設定すると共に全てのサーボドライバ内の位置フィードバック比例ゲイン K_{pp} を0とし、強制的に速度指令値を0とした速度フィードバック制御による減速処理を行わせるようすれば、位置指令や位置のフィードバックに如何なる値が来ようとも、速度指令を単純に0にしておくことができ、加えて、衝突による干渉力がトルク制限値を上回る場合、関節が干渉力を逃がす方向に動作して衝突力が軽減される。

【0176】ロボットの動作を停止させるために必要となる各関節のトルク値の算出においては、全ての関節が同じ時間で停止すると仮定し、関節ごとに減速停止時間を算出した後、全ての減速停止時間のうちの最大値をいずれの関節の減速停止時間にも充てるようすれば、衝

突しても逃げ動作する余地が残され、衝突による力を吸収して損傷の発生を可及的に少なく抑えることができる。

【図面の簡単な説明】

【図1】 本発明に係る多関節ロボットにおける衝突検出・停止制御法が適用されているロボット制御系全体を示したブロック線図。

【図2】 関節数を3として簡略化した構造のロボットを表すと共に、リンクが障害物に当たったときの逃げ動作を表したロボットの概略図。

【図3】 ロボットコントローラ（制御装置）とティーチペンダントならびに関節駆動用モータとの間の信号伝達の概要を示した制御ブロック図。

【図4】 衝突の有無を判定する処理手順を表したフローチャート。

【図5】 イナーシャテンソルの基準座標を与えた第2リンクの説明図。

【図6】 サーボドライバからコントローラへ伝達される応答値の変更例を含んだ制御系全体を示すブロック線図。

【図7】 トルク指令値に代えてトルク応答値を採用する場合のサーボドライバのブロック線図。

【図8】 ブラシレスDCモータを使用している場合のサーボドライバのブロック線図。

【図9】 軌道計画によって生成された関節位置から得られる速度・加速度をもとに逆運動力学演算させるようにしたロボット制御系全体を示すブロック線図。

【図10】 衝突の有無を判定するにおいて図9に基づく処理手順を採用した場合のフローチャート。

【図11】 速度をサーボドライバにフィードフォワードした場合のロボット制御系全体のブロック線図。

【図12】 速度指令に対する速度応答の遅れの程度を表したものであり、(a)は速度をフィードフォワードしない場合の応答図、(b)はフィードフォワードした場合の応答図。

【図13】 停止処理する手順を表したフローチャート。

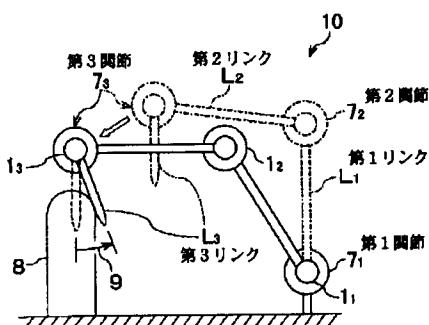
【図14】 モータを駆動するための各種制御ループが組み込まれたサーボドライバの構成概略図。

【図15】 サーボドライバから位置偏差を取り出して衝突の有無を判定すると共に、ひき続いて停止処理することもできるようになっていることを表した従来技術のブロック線図。

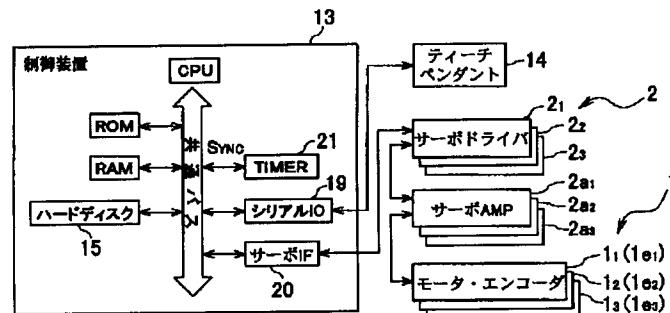
【符号の説明】

1, 1₁, 1₂, 1₃ …モータ、2, 2₁, 2₂, 2₃ …サーボドライバ、3…トルク制限器、7₁ …第1関節、7₂ …第2関節、7₃ …第3関節、8…障害物、10…ロボット、11…衝突検出部、12…停止処理部、13…コントローラ、15…ハードディスク、22…軌道計画部、23…電流検出器、25…速度・加速度計算部、29₁, 29₂, 29₃ …比較器、L₁ …第1リンク、L₂ …第2リンク、L₃ …第3リンク、θ_{m1}, θ_{m2}, θ_{m3} …サーボドライバが検出したモータの位置、θ_{d m1}, θ_{d m2}, θ_{d m3} …サーボドライバが検出したモータの速度、θ_{dd m1}, θ_{dd m2}, θ_{dd m3} …サーボドライバが検出したモータの加速度、T₁, T₂, T₃ …衝突が発生していない場合に各モータが発生しているトルク予測値、T_{m1}, T_{m2}, T_{m3} …サーボドライバが実際に生成したトルク指令値、ΔT₁, ΔT₂, ΔT₃ …トルク予測値とトルク指令値から算出されるトルクとの差分、ε₁, ε₂, ε₃ …衝突判定値、K_{pp}…位置フィードバック比例ゲイン。

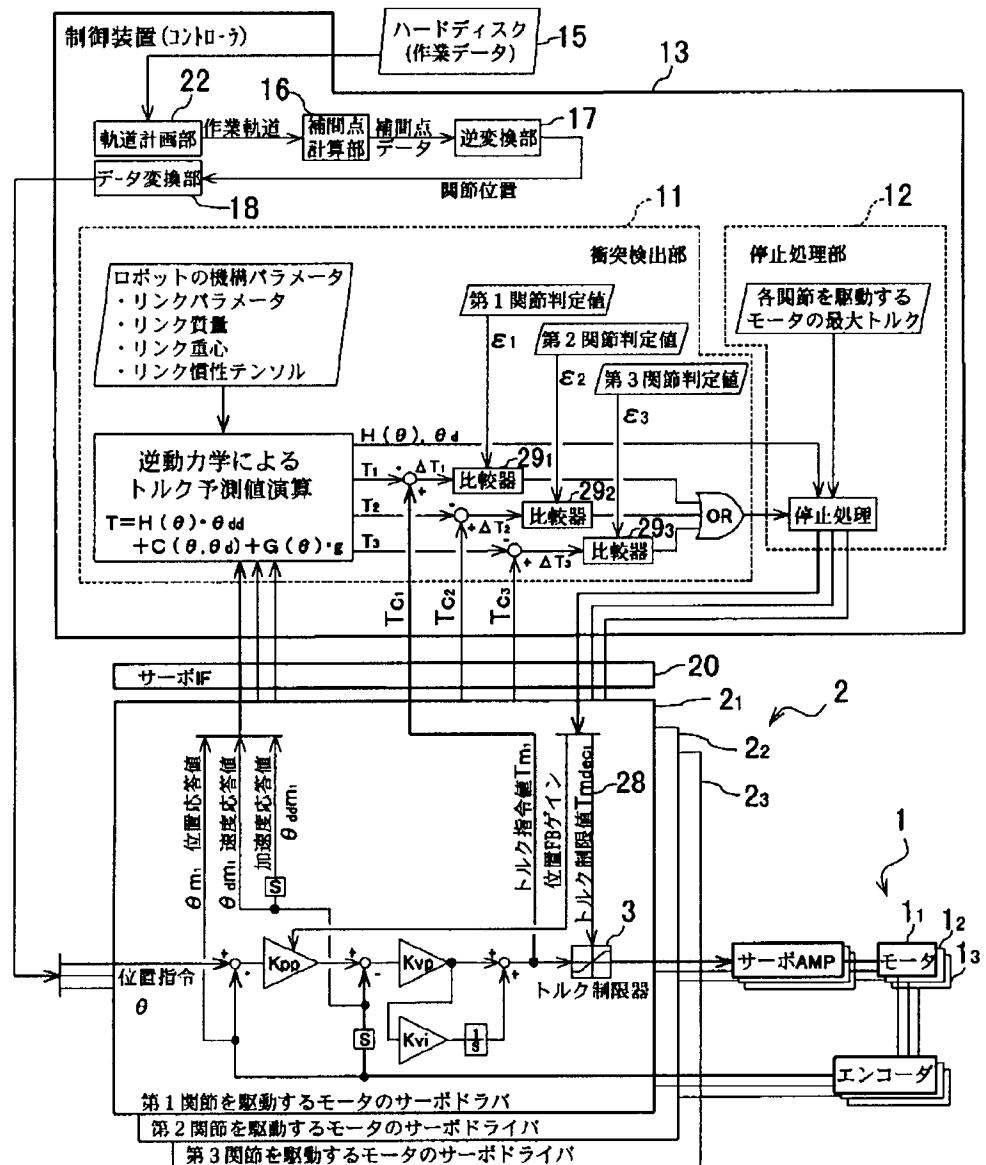
【図2】



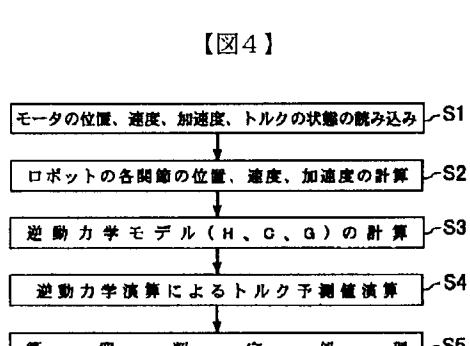
【図3】



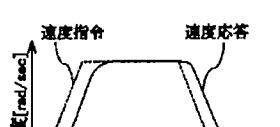
【図1】



【図4】



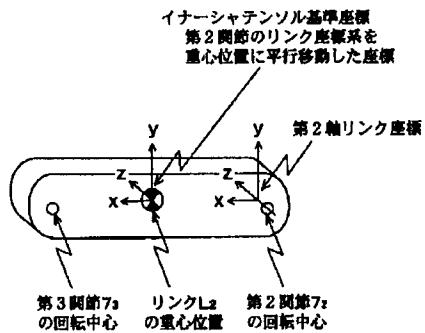
(a)



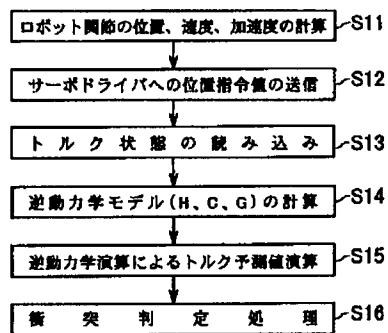
(b)



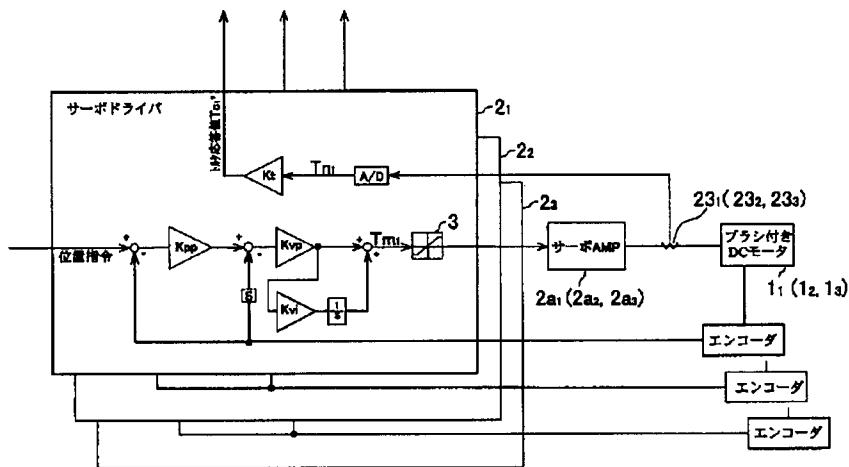
【図5】



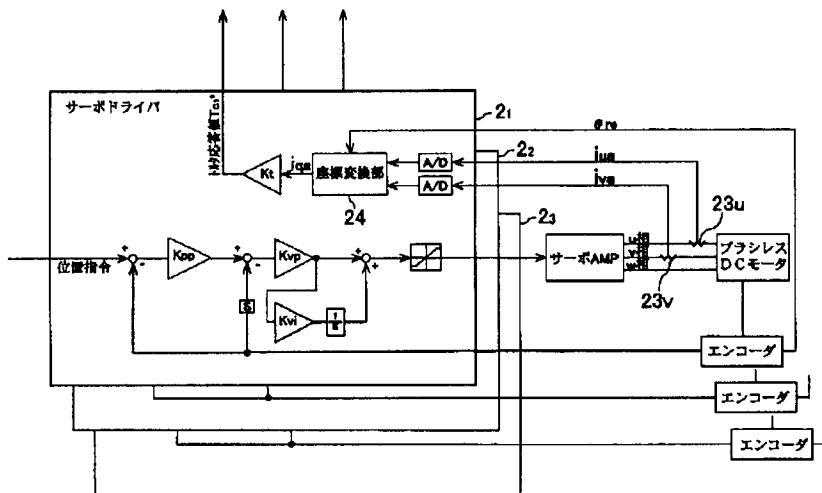
【図10】



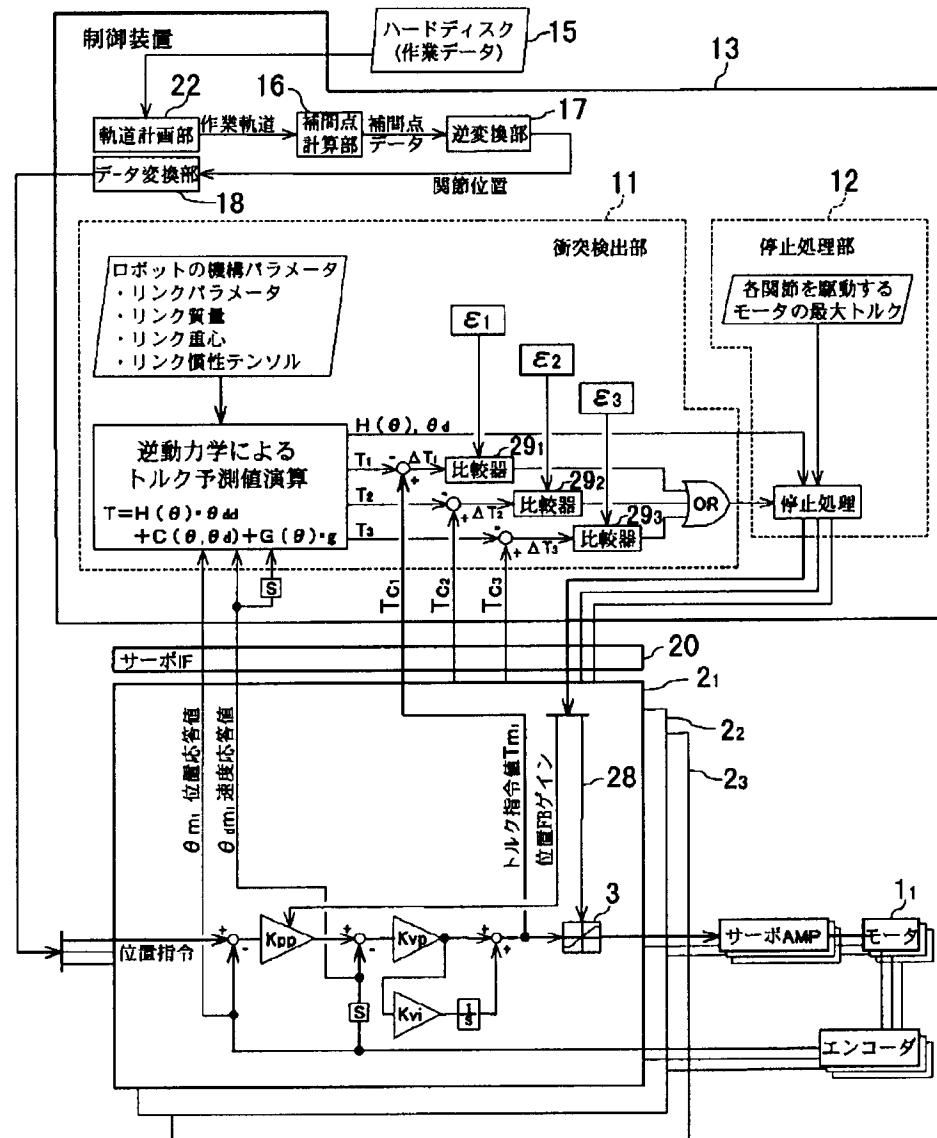
【図7】



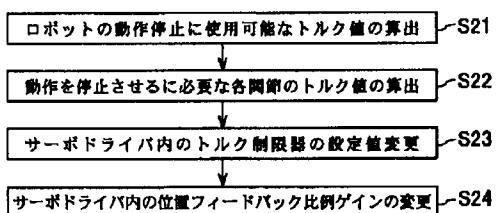
【図8】



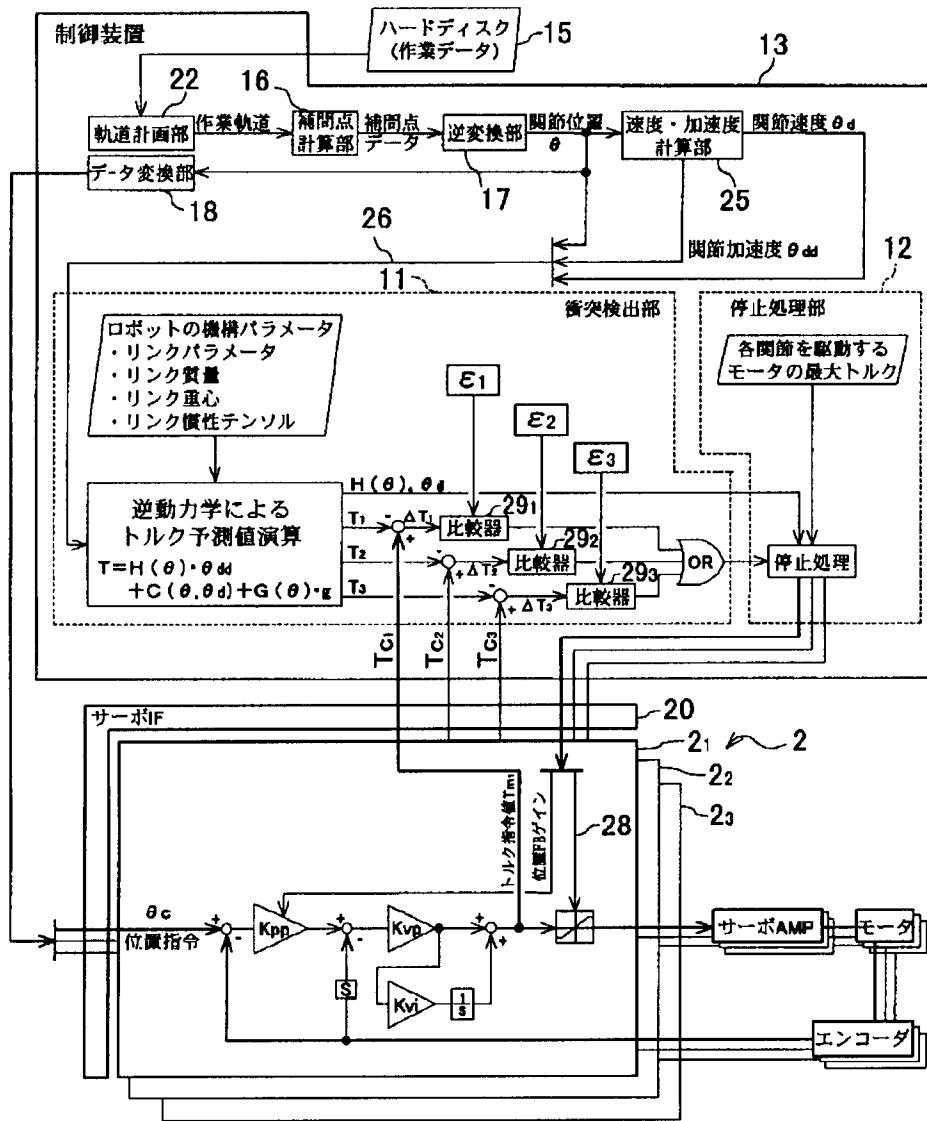
【図6】



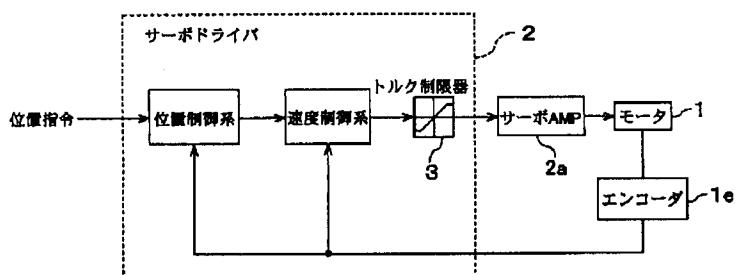
【図13】



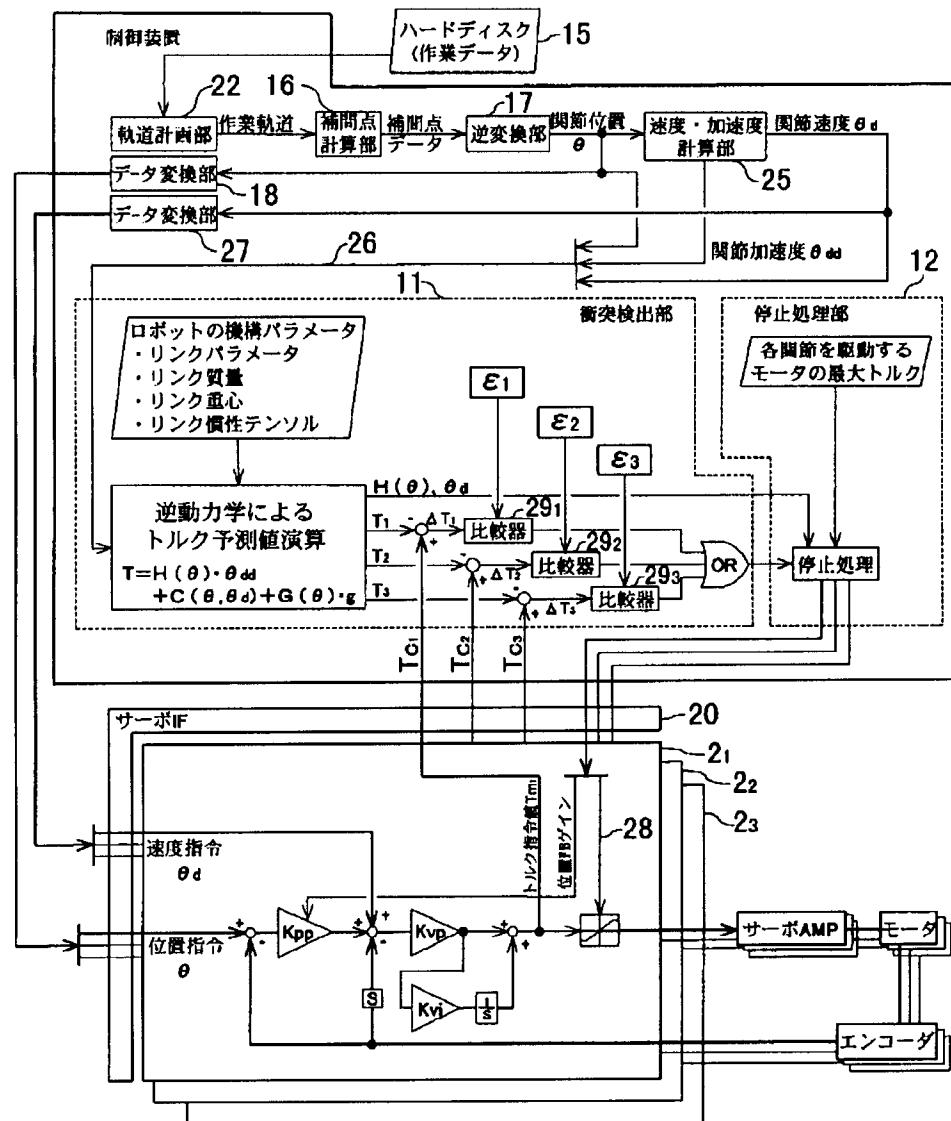
【図9】



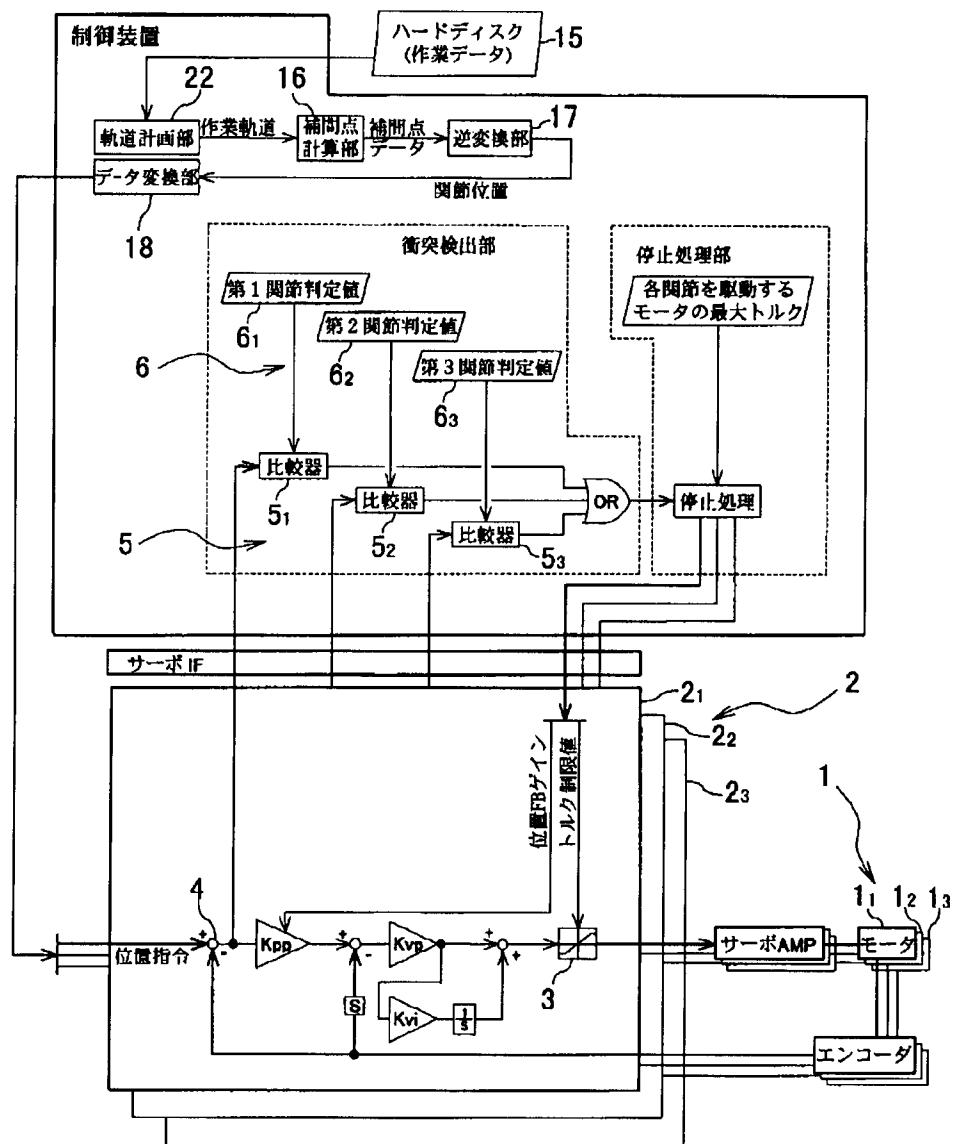
【図14】



【図11】



【図15】



フロントページの続き

F ターム(参考) 3C007 AS11 BS10 KS16 KS22 KS23

KS35 KS37 LU03 LV15 LV19

MS03 MS14

5H269 AB12 AB33 BB11 CC09 DD05

EE01 EE14 GG01 MM05 NN04

PP03